Department for Transport

TransXChange – An XML Standard for the Data Exchange of Bus Schedules and Related Information.

TransXChange Schema Guide

2.5

Department for
Transport

Version History

Schema	Version	Date		Review
2.0a	0.1 Preliminary Consultation Draft	03 04 2004	NJSK	Review
2.0b	0.4 Consultation Draft	10 03 2004	RM /NJSK	Review
2.0c	0.9 Consultation Draft	11 05 2004	NJSK	Review
2.0c	0.10 Consultation Draft Corrections	12 05 2004	/NJSK	Review
2.0c	0.15 Corrections, added dead run, track & revised operation date sections.	14 05 2004	NJSK	Internal
2.0d	0.16 Corrections.	09 06 2004	TW	Internal
2.0e	0.25 Clarifications & Corrections	15 08 2004	NJSK	Internal
2.0e	0.26 Minor formatting corrections	18 08 2004	NJSK	Review
2.0f	0.27 Add Public Use,	26 08 2004	NJSK	Review
2.0f	0.31 Corrections, renumber figures and tables, Add Booking Arrangements, Legislative references, Block, Refine integrity rules. Drop PPT	07 10 2004	NJSK	Internal
2.0g	0.32 Corrections. Revise Transmodel comparison, Refine integrity rules.	16 12 2004	NJSK	Review
2.0g	0.33 Corrections. From RS	23 01 2005	NJSK	Review
2.0g	0.34 Clarify MDV points	30 02 2005	NJSK	Review
2.0	0.35 Release 2.0 Clarify versioning points	10 03 2005	NJSK	Final
2.1	0.36 Release 2.1 Minor fixes and corrections	27 09 2005	NJSK	Issued
2.1	0.37 Minor fixes and corrections	06 01 2006	NJSK	Issued
2.1	0.38 Clarify use of duration	15 11 2006	NJSK	Review
2.1	0.39 Update with Publisher enhancements in route merge	16 03 2007	NJSK	Issued
2.1	0.40 Clarify route map, correct times, improve diagrams	14 07 2007	NJSK	Issued
2.2a	0.41 Clarify rounding,	24 07 2007	NJSK	Review
2.2a	0.41 correct subsidy classification	24 09 2007	NJSK	Review
2.2a	0.42 correct clarify use of journey times classification	12 03 2008	NJSK	Issued
2.2a	0.43 clarify stop request	03 09 2008	NJSK	Issued
2.2a	0.44 Revise all diagrams	02 03 2009	NJSK	Issued
2.2a	0.45 Remove wrong reference to change classification	10 06 2009	NJSK	Issued
2.2a	0.46 Update validation table	10 09 2009	NJSK	Review
2.4a	0.48 Update diagrams and 2.4a changes	16 02 2010	NJSK	Issued
2.4b	0.50 2.4b Corrections & PTIC Review feedback changes	16.06.2010	NJSK	Review
2.4	0.52 Update diagrams and 2.4	31 12 2010	NJSK	Issued
2.4	0.52 Clarify use of Address as per BS 7666	12.06.2012	NJSK	Issued
2.5a	0.54 Accessibility changes	28.03.2012	NJSK	Review
2.5b	0.55 Review comments, add ServiceFaciltySet	14.05.2012	NJSK	Review
2.5b	0.57 Correct spelling of Prrnciple timing point	10.09.2013	NJSK	Review
2.5	0.58 Minor corrections to diagrams and added guidance.	04.10.2013	NJSK	Review
2.5	0.59 Minor clarifications and issue.	08.01.2014	NJSK	Issued

Prepared By:

www.trapezegroup.co.uk

Nicholas Knowles
Email: mailto:schemer@transxchange.org.uk

Prepared For:

Department for Transport

T

2/17 Great Minster House,
33 Horseferry Road,
London, SW1P 4DR

transport direct.info
Connecting People to Places

CONTENTS

Section *Page*

List of Figures

List of Tables

1 INTRODUCTION

TransXChange is a UK national data standard for the exchange of bus route and timetable information. The standard is sponsored by the UK Department for Transport and is mandated by the Vehicle Operating Services Agency (VOSA) for the electronic registration of UK bus services with Traffic Area Offices (TAO) and Local Authorities.

TransXChange allows the exchange of route and timetable information between, amongst others:

- Bus Service Operators
- Traffic Area Offices
- Local Authorities
- Passenger Transport Executives
- Traveline – the National Passenger Transport Information System
- Suppliers of AVL (Automatic Vehicle Location) and delivery systems

TransXChange comprises a set of W3C XML schemas with related documentation and other tools.

This *Schema Guide* is intended to provide a technical overview and reference manual to *TransXChange* for system developers, data providers and other users of *TransXChange*.

The Guide is accompanied by a set of worked examples, available at the www.transxchange.org.uk web site. These provide explanations, diagrams and XML for using every feature of *TransXChange.* A summary table of the examples is given in Section 5.

Note that detailed documentation of individual schema elements is provided as annotations within the schema itself. Software Tools such as XML SPY can be used to explore the structure and details of the schema.

1.1 Antecedents

Version 1.0 of *TransXChange* was originally developed by Cap Gemini in 2001 for the Traffic Area Network (TAN) under contract to the UK Department for Transport. The *TransXChange* model for public transport schedules was based on *Transmodel*, the European standard reference Data Model for Public Transport. *Transmodel* is intended:

- To promote a common integrated approach in the design of public transport information systems.
- To provide an open architecture for such systems.
- To provide a general model that can easily be adapted to create specific implementations.
- To support the reliable exchange of information between different software products.

An early version of *Transmodel* provided a starting point for the *TransXChange* Logical Reference Model that underpins the *TransXChange* XML schema. As a comprehensive, supplier-neutral, general purpose information model for transport information, *Transmodel* provides a valuable overall context of concepts and terminology extending over most aspects of public transport information (see Section 13). However, it should be noted that *Transmodel* is an abstract model, and it covers a wider scope of function than that required for *TransXChange*. Furthermore, *Transmodel* was expressed primarily in terms of an Entity-Relationship model, without the benefits of the encapsulation and richer constraints available in an Object-based language such as XML. A concrete XML implementation such as *TransXChange* must make a specific interpretation of the subset of *Transmodel* that is salient for its objectives, and must use the data types and other capabilities of its technology. The main divergences from *Transmodel* terminology are listed in section 13.2.

Subsequent updates, also managed by Cap Gemini, developed revised releases 1.1, 1.2, & 1.2.1

TransXChange version 2.0, (2003-2004) was a major revision of the standard, managed by Carl Bro and Kizoom, which included harmonisation with government standards for XML schemas, and addressed a number of issues arising from early-adopters' experience of the initial version.

2.1 (2005) was a very minor update to 2.0 to harmonise with other changes to NaPTAN. All TransXChange 2.1 documents should be fully compatible with 2.1 tools for import.

2.2 (end 2008) was a minor update to 2.1 to enable the publishing of route maps and to demonstrate multi-level version support for the publisher.

2.4 (2010) is an extensive update with a number of new features as described later below. It includes extensive revisions to this document including a new set of UML diagrams. It has been remodularised internally to facility maintenance and further alignment with Transmodel. An updated version of the publisher is available.

2.5 (2013) is an minor update to 2.4 with a number of new features to support in partiucalr accessibility as described later below. It includes revisions to this document including an updated set of UML diagrams, recoloured to align with NeTEx's use of colours to indicate different functional groups of conceptual elements. An updated version of the publisher is required.

The TransXChange Publisher, a tool used to produce human readable timetables from TransXChange documents was provided with release 2.0 and onwards. A new enhanced version of the publisher was produced in 2008 including a desktop interface. This includes support for a multiple schema levels: the latest includes including 2.1, 2.2a and 2.4 versions of the schema.

1.2 Document Structure

The *TransXChange Concept Guide* is organised as follows:

Part I – Introduction & Overview.
The chapters in Part I are intended to give a summary of the basic concepts and purpose of *TransXChange*:

- Information about the *TransXChange Concept Guide*.
- The Purpose of *TransXChange.*
- *TransXChange* Basic Concepts.
- *TransXChange* Logical Model.

Part II – Worked Examples.
Part II provides an example of the components of a *TransXChange* document.

- Simple Worked Example.
- It also provides an index to the systematic set of examples demonstrating the use of all *TransXChange* features that may be found at the web site.

Part III – Schema Structure.
The chapters in Part III provide a detailed account of the *TransXChange Schema* elements:

- Topographical Elements: Stops & Localities.
- Network Elements: Routes & Tracks.
- Service Description Elements.
- Operational Date & Time Elements.

Part IV – Technical Reference.
The chapters in Part IV provide technical details on various aspects of *TransXChange* documents and technology:

- Technical Annexes.
  - Registration Process
  - *TransXChange Publisher.*

    - Naming and Coding Conventions.
    - *Transmodel* comparison.
    - Versioning.
    - Integrity Rules.
- Reference Appendixes.

1.3 Intellectual Property Rights

1.3.1 TransXChange Schema

TransXChange is Crown Copyright, managed by the UK Department for Transport. The schema may be used without charge.

The *TransXChange* Schema may reference other Schemas that are also Crown Copyright, or that are owned by Associate Members of the UK Government GovTalk initiative.

Anyone who wishes to reproduce the Schema in any format must acknowledge the source and state that the Schema are the copyright of the named Associate Member or Crown Copyright, as appropriate. The permission to reproduce does not extend to any Schema or parts of Schema which are specifically identified as being the copyright of anyone who is not a Member or Associate Member. Permission to reproduce these Schema or parts of these Schemas must be obtained from the identified copyright holders.

TransXChange is based on open source software standards, notably XML.

The designated owner of the *TransXChange* schema for GovTalk is:

TransXChange, Transport Direct
Department for Transport,
2/17 Great Minster House,
33 Horseferry Road,
London
SW1P 4DR

1.3.1.1 TransXChange Schedules

Rights in the contents of bus schedules encoded as *TransXChange* conformant XML documents are separate from rights in the *TransXChange* Schema itself. Document content is the property of the publisher of each document.

Data usage rights may be specified within a document for individual elements or for the whole document (+TXC 2.4). TransXChange 2.4 introduces tags for recording the data rights in a document. See later.

1.3.1.2 TransXChange Document Publisher

TransXChange includes a software tool, the *TransXChange Publisher*, which may be used to transform XML schedules into pdf output. *TransXChange Publisher*, is supplied on a free-to-use licence on an unwarranted, 'as-is' basis. The publisher runs under a Java environment (JRE 1.4.2 or higher).

The Publisher may use on-line web services to fetch stop and map data that are incorporated into the published output. The use of stop data and map data in published output is governed by the terms of use of the publisher. In particular the map data may only be used for validating TransXChange documents submitted to the EBSR process, and not for other commercial uses such as publicity material, planning, etc.

1.4 Versioning

A strict versioning system is used for *TransXChange*, following e-Gif principles. This was made explicit in Version 2.0 of TransXChange and is explained in Section 12.1.

1.5 Naming Conventions

Systematic Naming conventions are used for schema elements following the e-Gif guidelines. The conventions are described in Section 8.

1.6 Presentation Conventions Used in the Schema Guide

1.6.1 XML Elements in Text

TransXChange uses the XML Schema Language (See http://www.w3.org/TR/xmlschema-0/, http://www.w3.org/TR/xmlschema-1/ and http://www.w3.org/TR/xmlschema-2/) and its terminology, such as “sequence” and “choice” to formally describe its data structures.

Throughout this *TransXChange* Schema Guide:

- XML elements are shown in bold italic type, for example the ***JourneyPattern*** element.
- XML attributes are shown in bold, for example **MappingSystem**.
- Containment of a subelement by another element is shown by a forward slash, for example ***StopPoint*** / ***AtcoCode***.

1.6.1.1 UML Diagrams

Unified Modelling Language (UML) notation is used for class and instance diagrams to show the formal structure of the *TransXChange* Logical Reference model; the diagrams express structure in terms of classes, connected by association, aggregation and inheritance relationships, corresponding to the semantics available in XML’s built-in reference and extension mechanisms. Note that the UML diagrams are provided for explanatory purposes only, and omit an amount of detail (in particular, only a few element properties are typically shown as class attributes, and intermediary elements of a relationship are sometimes omitted.). UML notation uses well known conventions for showing the navigability, multiplicity, etc., of model elements, which we do not repeat here.
Note that in UML structure diagrams we label relationships in the direction of the navigability. Most relationships are navigable in only one direction, indicated by the arrow that points in the direction of navigability, i.e. coming from the entity that holds reference, to the referenced entity.
For *TransXChange, we* refine the standard UML conventions by the systematic use of colour: in particular:

- Network topology elements are shown in diagrams in *green* (for example, ***Route***, ***StopPoint***).
- Service level and service pattern related elements are shown in *yellow* (for example, ***FlexibleService***, ***JourneyPattern, JourneyPatternTimingLink***).
- Vehicle journey related elements are shown in *orange* (for example, ***VehicleJourney***, ***VehicleJourneyTimingLink***).
- Elements concerned with operational days, dates and times are shown in blue, (for example, ***OperatingProfile***, ***BankHolidays, Frequency***).

Different levels of detail are shown in the UML diagrams; introductory diagrams omit details and provide a high level overview; model diagrams show detailed attributes including physical attributes used to implement relationships; hierarchical views show the supertypes of objects; supporting diagrams show the low level data types used in the model diagrams.
Since we are depicting a physical model, in detailed diagrams we also indicate the attributes used to implement relationships.

1.6.1.2 XML Structure Diagrams

XML Spy (from Altova GmbH) structure diagrams are used extensively in the detailed schema description to illustrate the containment structure of XML schema fragments. Each XML element is shown as a solid box. Use of a complex data type is shown by a dashed box.
The presence of attributes is indicated by a '+. Since a common set of metadata attributes is used for first class objects, we do not generally show the attributes, though they may be listed in the accompanying documentation, using a convention of including the attribute name in the element comment prefixed by an 'at' sign ('@'), for example '@*lang*'.

1.6.1.3 Element Structure – Sequence

The hexagonal symbol with the horizontal line of three dots indicates "sequence of." For example, *Figure 1-1* says the element ***ValidityPeriod*** consists of the sequence of ***StartTime*** followed by ***EndTime***. Both elements are defined in the namespace whose prefix is "*txc*". The adornment of a small series of horizontal lines in their upper left box corners indicates that ***StartTime*** and ***EndTime*** have a simple type. Types are normally shown in the bottom half of the box.

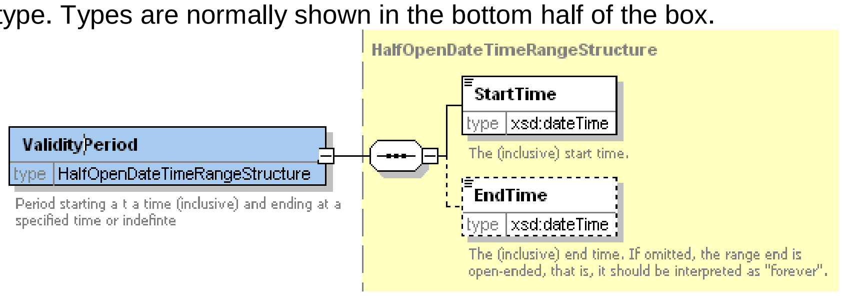


Figure 1-1 – XML Spy Diagram: Sequence

1.6.1.4 Element Structure – Choice

The hexagonal symbol with the switch-like icon indicates a choice. For example in *Figure 1-2* there is a choice between the elements ***NoSubsidy***, and ***Subsidy***. ***Subsidy*** has a further substructure, indicated by a "+" in at the right-hand end. ***NoSubsidy*** is simple type.

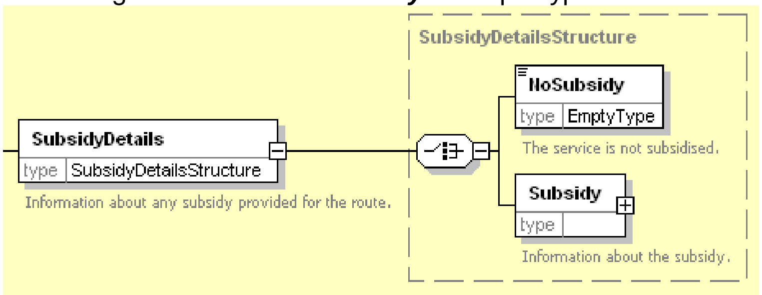


Figure 1-2 – XML Spy Diagram: Choice

1.6.1.5 Multiplicity and Optionality

Whether elements are required or optional, and the multiplicity (cardinality) of elements is indicated by adornments as follows:

- A *fine dashed line* on the connecting line and surrounding box indicates an element is optional. For example, in *Figure 1-3*; ***FlexibleZones*** and ***Description***.
- A *solid line* indicates a mandatory element. For example, in Figure 1-3; ***StopPointRef.***
- A *number adornment* indicates a multiplicity other than one. 'Many' is indicated by an infinity sign ∞. Thus, for example in *Figure 1-3*, there may be zero or one ***Activity instances*** per ***StopUsage,*** but there can be between one and many ***StopUsages*** per ***FlexibleZone***.

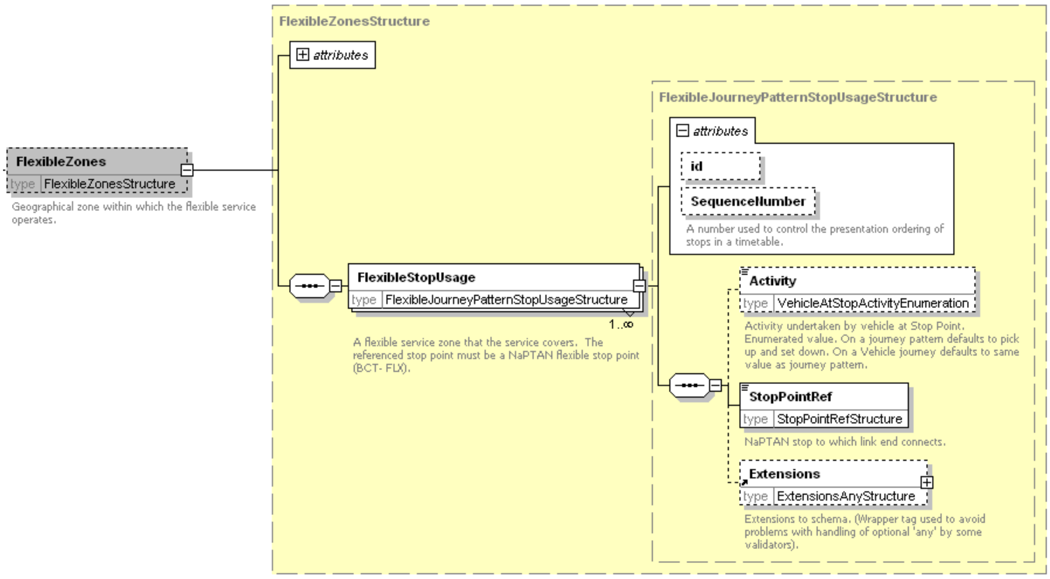


Figure 1-3 – XML Spy Diagram: Multiplicity

1.7 Summary of Changes

1.7.1 Changes in Release 2.5 of TransXChange

TransXChange 2.5 includes a small number of enhancements to improve support for accessibility and to integrate NaPTAN 2.5 changes to support Eire. In order to provide an underlying conceptual model and facilitate a long term harmonisation, the changes are based on NeTEx elements wherever possible.

- Functional
  - PTIC-083 NaPTAN change: Support for Eire locations on NaPTAN stops:
    - ITM (Irish Transverse Mercator) allowed as grid type.
    - Multiple Grid translations allowed.
  - PTIC-086 NaPTAN change: ***StopAccessibility*** added to StopPoint to allow stop level overrides of accessibility data.
  - PTIC-087 Accessibility Support
    - IFP 156 – Operator Booking Lines:
      - ***Operator*** extended with ***AccessibilityBooking*** information element.
    - IF145 Mode Accessibility:
      - Add new elements to ***WheelchairEquipment***, booking required, guide dog, etc.
      - Add AssistanceService to Operational group and to Service.
      - Add StopAccessibility to ***StopUsage***. Allows stop level overrides
      - Add Availability Condition to Equipment.
  - Align with JourneyWeb:
    - Allow a Transport Submode to be specified.
    - ***ServiceFacilitySet*** added to Operational data describe service properties such as fare classes, catering, toilets, aligned with JoruneyWeb. uSes a subset fo NeTEX.
- Documentation
  - Emphasize that journeys with the same operateor, line and journey pattern should be in the same service.
  - Emphasize that ***VehicleJourney*** values common to other journeys should be placed on the JourneyPattern and not on individual journeys.
  - Give guidance on populating ***StartDate*** and ***EndDate*** for Service ***OperatingPeriods***.

- Emphasize that **Special Days** should only be used for immediate exceptions.
- Emphasize that **RunTime** values should be given for a ***JourneyPatternTimingLinks*** in order to enable precise journey times.
- Emphasize that ***DeadRuns*** with **RunTime** values should be provided in order to facilitate accurate real-time calculation of service times.
-

- Technical
  - The version number attribute on a TransXChange document was previously a fixed value (e.g. 2.1, 2.4, etc.). It is now a variable that defaults to the current value (e.g. 2.5). This makes it easier for implementers to use a single schema binding with documents that conform to earlier releases.

1.7.2 Changes in Release 2.4 of TransXChange

TransXChange includes a number of semantic revisions to add in features requested by the PTIC TransXChange user group and to enable further harmonisation.

Functional

- PTIC-XXXX-Authority Changes Addition of ExeterCity & NorwichCity Areas, modularisation of Areas to a separate schema file.
- PTIC-001 Add Additional National Operator Database attributes to Operator.
- PTIC-002 Add Partial frequent Service Interval.
- PTIC-003 Relax constraints on service classification.
- PTIC-011 Temporal grouping of post-midnight journeys.
- PTIC-012 Explicit Journey Grouping.
- PTIC-013 Line Description by Direction
- PTIC-016 Additional business rule validation.
- PTIC-018 Support concise cancellation.
- PTIC-022 Footnote publishing & Serviced Organisations.
- PTIC-027 Multiple Operational References per Journey.
- PTIC-028 Add workflow attributes. *
- PTIC-029 Vehicle Accessibility info.
- PTIC-031 Permission Levels & IPR Use *
- PTIC-032 Support Dynamic Vias on StopUsage.
- PTIC-033 Add Recommended End date to Service Operating Period.
- PTIC-035 Support for general school holidays
- PTIC-036 Support Minimum layover time on Layovers.
- PTIC-037 Support DutyCrewCode on PositioningLinks.
- PTIC-038 Add CommercialBasis flags to Service, etc.
- PTIC039 Improve support for JourneyInterchanges.
- PTIC-040 Support for Jan2ndDisplacementHoliday & StAndrewsDay.
- PTIC-041 Support for Line Colours.
- PTIC-042 Add Marketing Name to Service Description.
- PTIC-044 Additional Change Management Support – Deltas: New Schema variant.
- PTIC-067 Permit use of & in Service Codes.
- PTIC-071 National Term database Support. +
- PTIC-074 Extensible Authority names

Technical

- T-001 Modularisation of TransXChange_common into discrete modules for each TransXChange component (JourneyPattern, operator etc.) to facilitate.
- T-002 Revision of all UML diagrams to use unified EA diagrams.
- T-003 Revision of all XML diagrams to show element types.
- T-004 Addition of Package modularisation diagrams.
- T-005 XML Refactoring of all ordinary objects as VersionedObject descendants. Addition of substitution groups to show hierarchy.

Elements that are added in v2.4 are marked (+TXC 2.4). Elements that are changed are marked 'Changed in v2.4'.

1.7.3 Changes in Release 2.2a of TransXChange.

TransXChange 2.2a included minor semantic revisions to enable the support of route maps.

- Addition of location to annotated Stop Refs.

1.7.4 Major Changes in Release 2.0 of TransXChange

TransXChange 2.0 included major syntactic and semantic revisions to bring it closer to *NaPTAN* and other standards. The following is a summary of major changes in release 2.0. See Section **Error! Reference source not found.** for a full list of changes.

- Modularisation.
- eGif GovTalk compliance.
- Data Integrity improved.
- Welsh Language support added.
- Route Links remodelled.
- VehicleJourney & JourneyPattern model revised for efficiency and integrity.
- Days of Operation standardised and extended.
- Registration Number supported.
- Provision of a full *TransXChange Schema Guide* with examples.
- New *TransXChange Publisher* to transform XML documents to Acrobat pdf format.
- Use of revised *NaPTAN* & *NPTG* models.
- Revision of Registration / Service relationship to enable connecting services to be specified in registrations.

New function for:

- New National Operator code, when available.
- Flexibly Routed Services.
- Vehicle Operations.
- School Dates.
- Fare Stages (but not fares).
- Dead Run support.
- Dynamic Bay Allocation.
- Add further descriptive elements to Service.

For changes in 2.1 see Appendix B.

Note that an extension of *TransXChange* to handle fares information, currently referred as *FareXChange,* is being considered for future development.

1.8 Evolving TransXChange

Successive versions of TransXChange introduce new features that are not present in previous versions: and documents containing these features may not be processed by tools that are designed for a previous version. However a principle of *full upwards compatibility of data* is upheld – the existing content used to create *TransXChange* documents should be exactly mappable to the revised schema at a new release. Since existing *TransXChange* documents are generated automatically by various suppliers' tools, the enhancement of the tools to generate the new format should provide a straightforward upward migration path.

2.0 put into place a formal versioning method that allows concurrent operation of schemas at different levels.

Normally strict compatibility is achieved in each TransXChange release, that is, if the new feature is not used in the document, the document may be processed as if it was of an earlier version.

1.9 Acknowledgments

The first version document was been prepared as part of the TransXChange 2.0 release and updated for 2.1 by the Carlbro (Richard Mejia, Paul Robinson) and Kizoom teams (Nick Knowles, Tom White) under direction of Roger Slevin of the Department for Transport, and Tim Hughes (VOSA). Introduction, modelling, structure example, schema and technical sections have been provided by Kizoom, worked examples by Carlbro. We thank Matt Francis of Action Information Management Ltd for his examples, comments and suggestions including the table of comparative terminology. Thanks also to Andrew Cudbertson (Arriva), Ross Dixon (CGEY), Michael Forbes (Opcom), Kieran Holmes (Cap Gemini), Paul Houghton (Trandata), Peter Miller (ACIS), Peter Neil (Trapeze), Mike Ness (WSAtkins), Pete Ridley (Thales), John Prince (SYPTE), John Gallagher (Thales), Stephen Corlett (Thales), Richard Shaw (WSAtkins), Alex Worrel (AtkinsGlobal), Adrian Walters (Infocell), Mary Doonan (Journey Plan), Dave Walter (Anite), Dr Martin Siczkowski (WYPTE), Mike James (Tandata), John Pryer (Omnibus), Wilfred Düx (MDV), Graham Browne (WYPTE), Peter Stoner, and other ATCO,RTIG and PTIC members for their comments, examples and other feedback.

The 2.4 version document was been prepared by a Kizoom team (Nick Knowles, Chris Anderson) for Centaur (Mark Cartwright) under direction of Chris Gibbard & Roger Slevin of the Department for Transport, The PTIC user group has contributed the ideas and priorities for the new features over the past several years. The enhanced National Operator Code Model draws on papers by Mark Fell and others.

1.10 Related Transport Information Standards

TransXChange is an XML based standard and is compatible with the following standards for public transport information:

- **ATCO-CIF:** ATCO-CIF is a general purpose exchange format for common elements of timetable information. *TransXChange* is the successor to ATCO-CIF.

- **NaPTAN:** The **National Public Transport Access Nodes** database is a UK nationwide system for uniquely identifying all the points of access to public transport in the UK. The *NaPTAN* database is maintained centrally under contract to the Department for Transport. The *NaPTAN* standard is described in a separate document (see bibliography at end). *NaPTAN* is intended to assign every UK train station, coach terminus, airport, ferry terminal, bus stop, etc, a unique *NaPTAN* identifier. For large interchanges & termini, *NaPTAN* points identify the entrances from the public thoroughfare – one identifier is distinguished as the main entrance.

- **NPTG:** The **National Public Transport Gazetteer** is an auxiliary database to *NaPTAN* that provides a means of relating *NaPTAN* stops to UK towns and villages, as well as to the regional groupings used to manage Public Transport data. *TransXChange* assumes knowledge of the current NPTG database by all parties.

- **Transmodel:** *Transmodel* is an abstract Reference Data model of the data of interest to organisations designing transport related information systems. It has been developed through several European Commission sponsored projects. An XML version is currently under development as NeTEx (Network Exchange). Changes to TransXChange are designed to conform to NeTEx data structures.

- **JourneyWeb:** *JourneyWeb* is an XML protocol allowing distributed journey planning. The protocol is a UK national *de facto* standard sponsored by the UK Department for Transport, and is being used in the Transport Direct Portal project to provide contiguous distributed journey planning across the whole of the UK.

- **SIRI:** *Service Interface for Real-time Information* is a standard for the exchange of real time bus information between systems developed by CEN members of the UK Real Time Interest Group. It is also based on *NaPTAN* and *Transmodel,* and will be evolved so as to harmonise with other related standards including *TransXChange*.

- UK **Geocoding** References: For geospatial location references *TransXChange* supports both Grid references – using Eastings and Northings, with support for both UK Mainland and Irish grids – and WGS 84 Latitude and Longitude. However Grid location references must be used for registrations.

1.11 Legislation

Bus registration is covered by several sets of bus registration regulations under the Transport Act 1985. These regulations are: The Public Service Vehicles (Registration of Local Services) Regulations 1986, amended by SI1988 1697, SI1989 1064, SI1993 2752, SI1994 3271 and SI2004 10.

1.12 Related Documents

A TransXChange Registration provides an electronic representation of the following forms issued by the Vehicle and Operating Services Agency (VOSA). The forms may be downloaded in pdf format from http://www.vosa.gov.uk/.

Description	England and Wales	Scotland	Date

Application to Register a Bus Service	PSV350	PSV350 (Scotland)	June 2003
Short Notice Registration Supplementary Form	PSV350A	PSV350A (Scotland)	Sept 2001
Local Bus Service Registration. Guide for Operators	PSV353A		June 2004
Application to Change or Cancel details of a Local Service Registration	PSV355	PSV355A (Scotland)	

Table 1-1 – Forms for Registering Bus Services in England and Scotland

2 OVERVIEW OF TRANSXCHANGE

2.1 The Purpose of TransXChange

TransXChange is a standard format for describing bus routes and schedules as XML documents that can be automatically exported and imported between different computer systems. The documents themselves can be exchanged by different transport mechanisms, for example, FTP, email or http, and can be zipped (compressed) significantly to speed transfer.

There are two main variants of *TransXChange*:

- **Registration schema:** Defines an XML document specifically for the purpose of registering bus services with VOSA. Each document contains a single registered "service". Includes integrity constraints to ensure the document elements are complete and consistent.
  - http://www.transxchange.org.uk/schema/2.5/TransXchange_registration.xsd.
- **General schema**: Defines an XML document for exchanging bus timetables and related information for many different purposes. More than one bus service can be specified in a single document. Includes integrity constraints to ensure the document elements are complete and consistent. May be used to exchange particular elements by themselves.
  - http://www.transxchange.org.uk/schema/2.5/TransXchange_general.xsd.

In addition there is a third variant that can be used just for exchanging just the updates to a previously exchanged document:

- **Delta schema**: Defines an XML document for exchanging just the changes to a previously exchanged bus timetables or related information. The schema does ***not*** have integrity constraints to ensure the document elements are complete and consistent
  - .http://www.transxchange.org.uk/schema/2.5/TransXchange_genera_delta.xsd.

2.2 TransXChange Components

TransXChange comprises the following components:

- *TransXChange Schema*: A model and formal XML schema (and variants) for describing and encoding bus schedules as XML documents. The schema can be used with software tools to check that documents are correctly formatted and contain the required content.

- *TransXChange Documents and Process*: A description and explanation of the standard, including rules for creating, managing and using *TransXChange* documents with software tools.

- *TransXChange Publisher*. The publisher is a free tool issued along with the *TransXChange,* schemas which allows users to render *TransXChange* XML documents into a readable timetable-like layout, using an Acrobat pdf or html output file format. The free Acrobat reader from Adobe can be used to read and print pdf files. *TransXChange Publisher* requires the installation of a standard open source environment for running Java and XSLT – this can also be downloaded free. Use of these tools is described in Chapter 9. The *TransXChange Publisher* can be run in two modes: for *Registration*, in which case a specific subset of content is published for the registered particulars of a service and for *General* Use, which includes some additional content.

It should be emphasised that *TransXChange* is a data definition standard, and not a software program or a dynamic protocol in itself. It is intended to enable different suppliers and user communities to build systems that can share information correctly, cheaply and efficiently, but does not prescribe detailed error handling or other implementation details – such as the exact representational model - that will vary according to the requirements of individual applications.

Different applications may use different proprietary internal representations to store timetables exchanged with TransXChange.

2.3 Document Validation

To be valid *TransXChange* data, documents must satisfy two levels of validity criteria:

1. **Well-formedness and validity:** Documents must parse and validate against the *TransXChange* schema at the specified level – **Registration** or **General** – including all the integrity constraints coded within the schema, such as for key uniqueness and reference. Any document that does not satisfy the syntactic rules will be rejected for Registration and is very unlikely to be accepted or understood correctly in uses under the General Schema.

2. **Correctness:** Documents must satisfy additional processing rules and constraints that are not enforceable in the XML of the schema, but which are specified in this document, or as annotations in the schema (In case of any inconsistency, the schema should be regarded as definitive). Typically these rules cover additional complex processing or uniqueness constraints that cannot readily be expressed using XML's built-in mechanisms. Any document that is not correct may be rejected for Registration and may not be accepted or understood correctly in uses on the General Schema. A number of semantic rules are listed later, and a severity assigned to them. The publisher provides a diagnostic function to checks for a number of these errors.

2.4 How is TransXChange Used?

The following three scenarios give the most common uses for *TransXChange*:

(i) To register a complete service.
(ii) To update a registration.
(iii) To exchange service related data for a wide variety of other purposes.

2.4.1 Registration of a Route with VOSA for EBSR

The most common scenario for use of *TransXChange* is to make a registration (*Figure 2-1*), with VOSA under the Electronic Bus Schedule Registration system and runs as follows:

1. Bus schedule data is prepared using scheduling software, including stop data from NaPTAN and route and geospatial data from other sources.
2. The schedule is exported as a *TransXChange* XML document to VOSA for registration. On export, the document is validated against a specified version of the schema. Note that *TransXChange* documents can also in principle be created by hand, though this would be both tedious and error prone.
3. The schedule is then imported by VOSA and Local Transport Authorities. On import, the document is validated against the version of the schema indicated by the document.
4. Following validation, the registered particulars alone are rendered as a readable pdf document using the Registration option of the *TransXChange* publisher.
5. The schedule is then imported by information system builders such as journey planners and AVL system implementers.
6. All or part of routes and schedules may be exchanged by system providers, annotated with additional operational data, over and above the registered particulars.

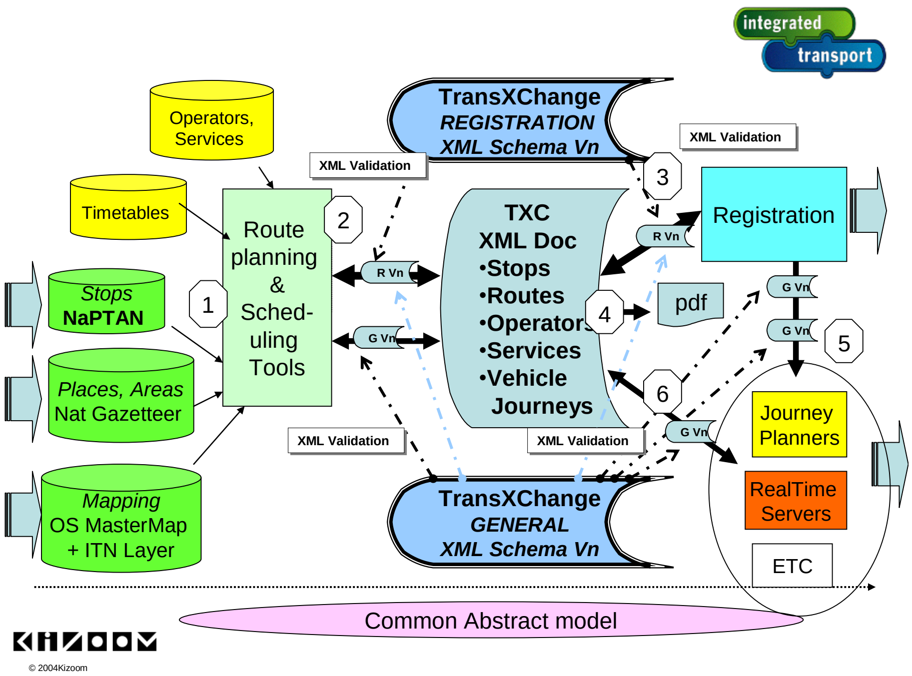


Figure 2-1 – Overview of TransXChange Use

2.4.2 Update of a Registration with VOSA

TransXChange will also be commonly used to update an existing registration.

1. The schedule is updated by the owner using the schedule preparation system.
2. The schedule is reported as an XML registration document with updated data and modified change dates. Note that the whole schedule must be recreated; *TransXChange* does not

currently formally support the exchange of 'deltas', that is, changes to just part of a route or timetable (though this is likely to be added in future).

3. The schedule is revalidated and imported by VOSA, and the changed parts are updated in the VOSA database. The validation and propagation process thereafter is as for registration.

2.4.3 General Purpose Exchange of Data

TransXChange can also be used for the general purpose exchange of structured bus schedule data between any two information systems. Normally the *TransXChange G*eneral schema will be used for this purpose, as it allows consistent subsets of data to be exchanged. Example uses might include:

- Exchanging schedule information with journey planning systems that wish to use the service.
- Exchanging route information with mapping systems that wish to draw the route.
- Exchanging schedule and operational data with AVL systems that wish to provide real-time bus predictions.
- Exchanging school term dates with Educational Authorities.
- Exchanging Operator details.

The precise scenario of use will depend on each specific purpose, but may be described generally (*Figure 2-1*), as follows:

- The exporting system will output the desired selection of data into an XML document. The resulting document must validate against the *TransXChange* schema version referenced in the document header.
- The document is transferred from the source to the target system by any appropriate transport method (e.g. email, ftp, and http).
- The importing system validates and imports the document, using the appropriate version of the *TransXChange* schema indicated by the document to interpret the document's contents. It will reject the document if it is not well-formed (including the rules for internal integrity). It may decide its own actions to handling errors in the conforming to application level integrity constraints.

2.4.4 General Purpose Exchange of Data changes

A variant of the *TransXChange* general schema can be used to exchange *'deltas'* just the changes since a previous exchange of data. *A* baseline version number and a "Changes since" date value can be used to indicate the relation to the previous release:

2.5 Differences between the Schemas

The *TransXChange* ***Registration*** and ***General*** schema are essentially the same, but differ in a few constraints as to cardinality and the required use of certain elements.

Table 2-1 summarises the differences between the two schema variations:

TransXChange Registration Document	*TransXChange* General Document
Must have a single ***Registration***.	Can have zero or multiple *Registration* instances.
The ***Registration***. Must reference a primary ***Service*** which describes the service being registered. Other connecting services to which the primary service connects can be included.	May have zero, one or many services.
The ***Service*** for the ***Registration*** must have a fully completed Registered Operator, i.e. of type ***LicensedOperator***.	Registered operator details need not all be completed, i.e. can be of type ***Operator,*** rather than ***LicensedOperator.***
Sufficient information about each stop must be provided to constitute a stand-alone definition for statutory purposes.	Simple Stop references may be used.
The ***Route*** information for the registered ***Service*** should include additional mapping point information where appropriate (using the ***RouteLink*** / ***Track*** / ***Mapping*** elements) to make the route unambiguous when the stop and mapping points are followed in sequence over a map containing a road network description.	Mapping information is optional.
Primary ***LocationSystem*** used in a ***Registration*** document must be Grid.	Either WGS84 or Grid can be used for ***LocationSystem***. The same system should be used for all references in a given document.

Table 2-1 – Differences between Schemas

The schemas share a common set of element types (*Figure 2-2*). As a general principle, the *Registration* schema is strictly substitutable with the *General* schema, that is, a valid *Registration* document will always validate against both schemas and can be used wherever a *General* document is used.

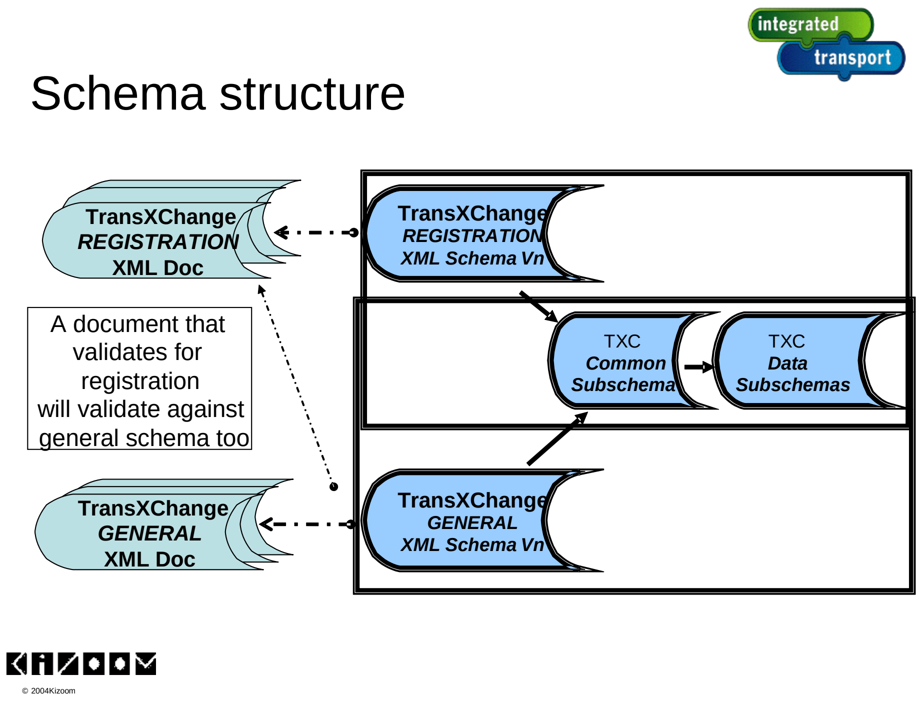


Figure 2-2 – Common Set of Types in TransXChange Schemas

3 SHORT TOUR OF THE *TRANSXCHANGE* ESSENTIAL MODEL

In this chapter, we provide an overview of the physical model underlying the *TransXChange* schemas. Unified Modelling Language (UML) diagrams are used to show the relationships between the most significant elements.

3.1 Representing a Bus Service in *TransXChange*

The *TransXChange* model) has seven basic concepts: ***Service***, ***Registration***, ***Operator***, ***Route***, ***StopPoint***, ***JourneyPattern,*** and ***VehicleJourney***.

- A ***Service*** brings together the information about a registered bus service, and may contain two types of component service: *Standard* or *Flexible*; a mix of both types is allowed within a single ***Service***.
- A normal bus schedule is described by a ***StandardService*** and a ***Route.*** A ***Route*** describes the physical path taken by buses on the service as a set of route links.
- A ***FlexibleService*** describes a bus service that does not have a fixed route, but only a catchment area or a few variable stops with no prescribed pattern of use.
- A ***StandardService*** has one or more ***JourneyPattern*** elements to describe the common logical path of traversal of the stops of the ***Route*** as a sequence of timing links (see later), and one or more ***VehicleJourney*** elements, which describe individual scheduled journeys by buses over the ***Route*** and ***JourneyPattern*** at a specific time.
- Both types of service have a *registered* ***Operator,*** who runs the service. Other *associated* operator roles can also be specified.
- **Route**, **JourneyPattern** and **VehicleJoumey** follow a sequence of *NaPTAN* ***StopPoints***. A ***Route*** specifies in effect an ordered list of ***StopPoints.*** A ***JourneyPattern*** specifies an ordered list of links between these points, giving relative times between each stop; a ***VehicleJourney*** follows the same list of stops at specific absolute passing times. (The detailed timing Link and elements that connect ***VehicleJourneys, JourneyPatterns*** etc to ***StopPoints*** are not shown in *Figure 3-1*). ***StopPoints*** may be grouped within ***StopAreas.***
- The ***StopPoints*** used in a **JourneyPattern** or ***Route*** are either declared locally or by referenced to an external definition using an ***AnnotatedStopRef***
- A ***Registration*** specifies the registration details for a service. It is mandatory in the registration schema.

Figure 3-1 introduces, in UML class diagram notation, the core elements of the *TransXChange* schema. Reusable elements with a global scope are organized beneath the root ***TransXChange.***

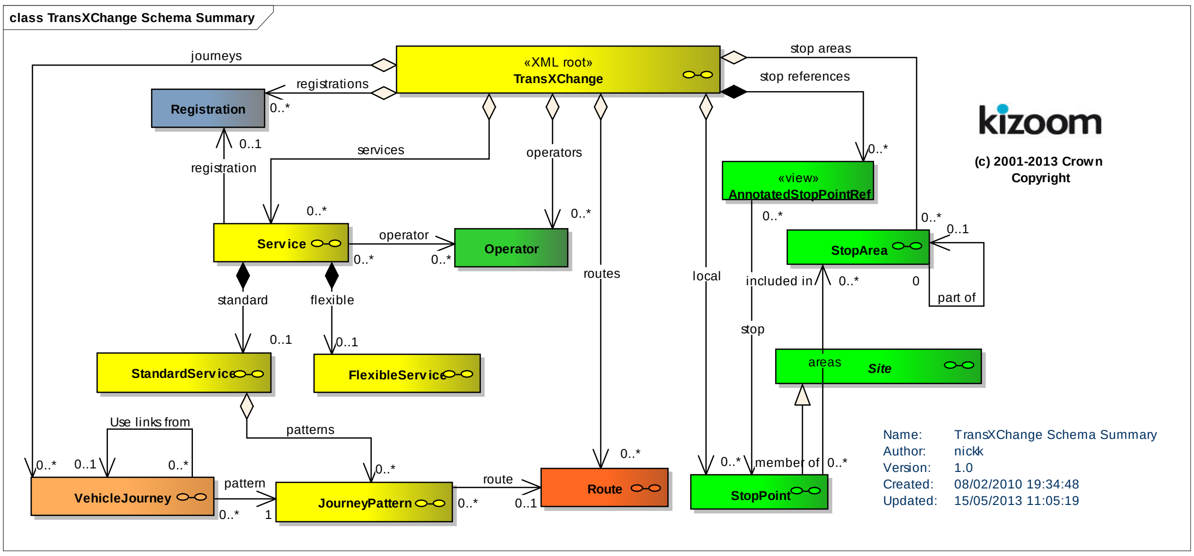


Figure 3-1 – UML Overview of TransXChange Model for a StandardService

Figure 3-2 shows further elements of the TransXChange model.

- A ***Calendar*** may be specified that defines ***OperatingDays***, ***DayTypes***, and assignments between them.
- A **ServicedOrganisation** can be used to specify a school, works or other organisation served by a ***Service***.
- A ***StopPoint*** may be part of a group of stops making up a ***StopArea***, and may reside in a topographic region specified by an ***NptgLocality***. Localities may be declared locally, or by reference to an external definition using an ***AnnotatedNptgLocalityRef***
- A ***Route*** may be made up of reusable ***RouteSections.***
- A ***JourneyPattern*** may be made up of reusable ***JourneyPatternSections***.
- A ***Registration*** may be accompanied by ***SupportingDocuments*** that pertain to it. Other ***SupportingDocuments*** may also be associated with the document as a whole.
- ***DataRights*** may be specified for the use of data elements.

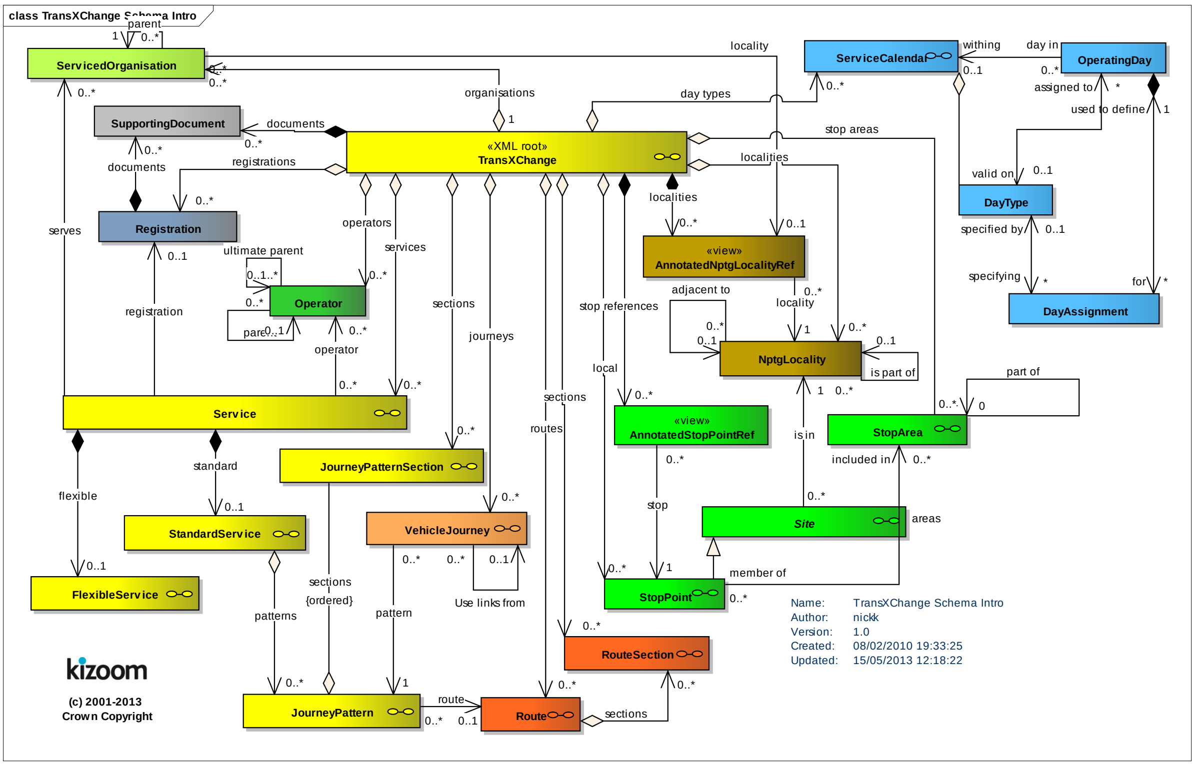


Figure 3-2 – UML Diagram of Elaboration of TransXChange model

3.1.1 The NaPTAN Stop Model

TransXChange uses the *NaPTAN* stop model to define the stops and timing points of routes, and to associate stops with topographical locations in the National Public Transport Gazetteer (*NPTG*). For further details refer to the '*NPTG and NaPTAN Schema Guid*e'.

Normally in *TransXChange,* stops comprise just a reference to an existing *NaPTAN* definition using a stop code; all such references are declared as ***AnnotatedStopPointRef*** instances. However, full ***StopPoint*** definitions for new bus stops may also be provided locally in a *TransXChange* document, using the *NaPTAN* ***StopPoint*** elements within the document. Each new locally defined stop definition must be allocated a *NaPTAN* identifier (that is an ***AtcoCode***) that can be used to reconcile them with the *NaPTAN* database later.

3.1.1.1 The NaPTAN Stop Model Introduction

Figure 3-3 summarises, in UML class diagram notation, the main stop elements of the *TransXChange* schema.

Stops are described using three main elements:

- ***StopPoint***: Describes a stop, it contains a place, which is used to associate the stop with an ***NptgLocality:*** localities are defined in the *NPTG* database and are open to the Local Transport Authority to edit. Stops may be of a number of different types and subtypes, each with different properties.
    - ***OnStreet / Bus:***
        - ***MarkedStop.*** UnmarkedStop***, HailAndRideSection, FlexibleZone.***
    - ***OffStreet /*** BusAndCoach
        - ***Bay, VariableBay.***
- ***StopArea***: Used to group stops together.
- ***NptgLocality***: Representing a topographical locality in the country, such as a city, town or village. Localities must exist in the *NPTG* database. Used to specify where a ***StopPoint*** or ***StopArea*** is relative to towns and cities.
- ***AdministrativeArea***. All NaPTAN and NPTG elements are assigned to an administrative areas – this represents the organisation responsible for maintaining the stop data. See NaPTAN schema guide for further details.

StopPoints may be declared as either a ***StopPoint,*** or ***AnnotatedStopPointRef,*** indicating that further details may be found in the NaPTAN database. The latter is the normal mechanism.

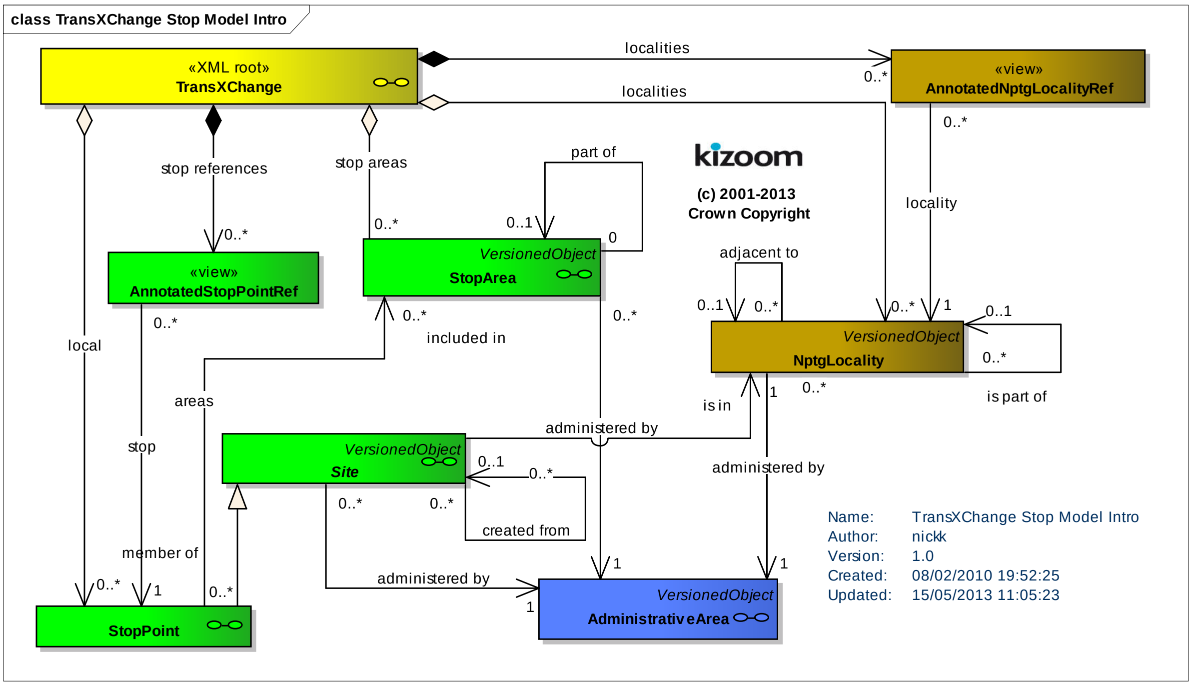


Figure 3-3 – UML Diagram of Summary of Stop Model

3.1.1.2 The NaPTAN Stop Model Details

Figure 3-4 shows further details of the NaPTAN stop elements. A ***StopPoint*** definition includes a ***Place*** & ***Descriptor*** groups. A ***StopAvailability*** may specify when a stop is available.

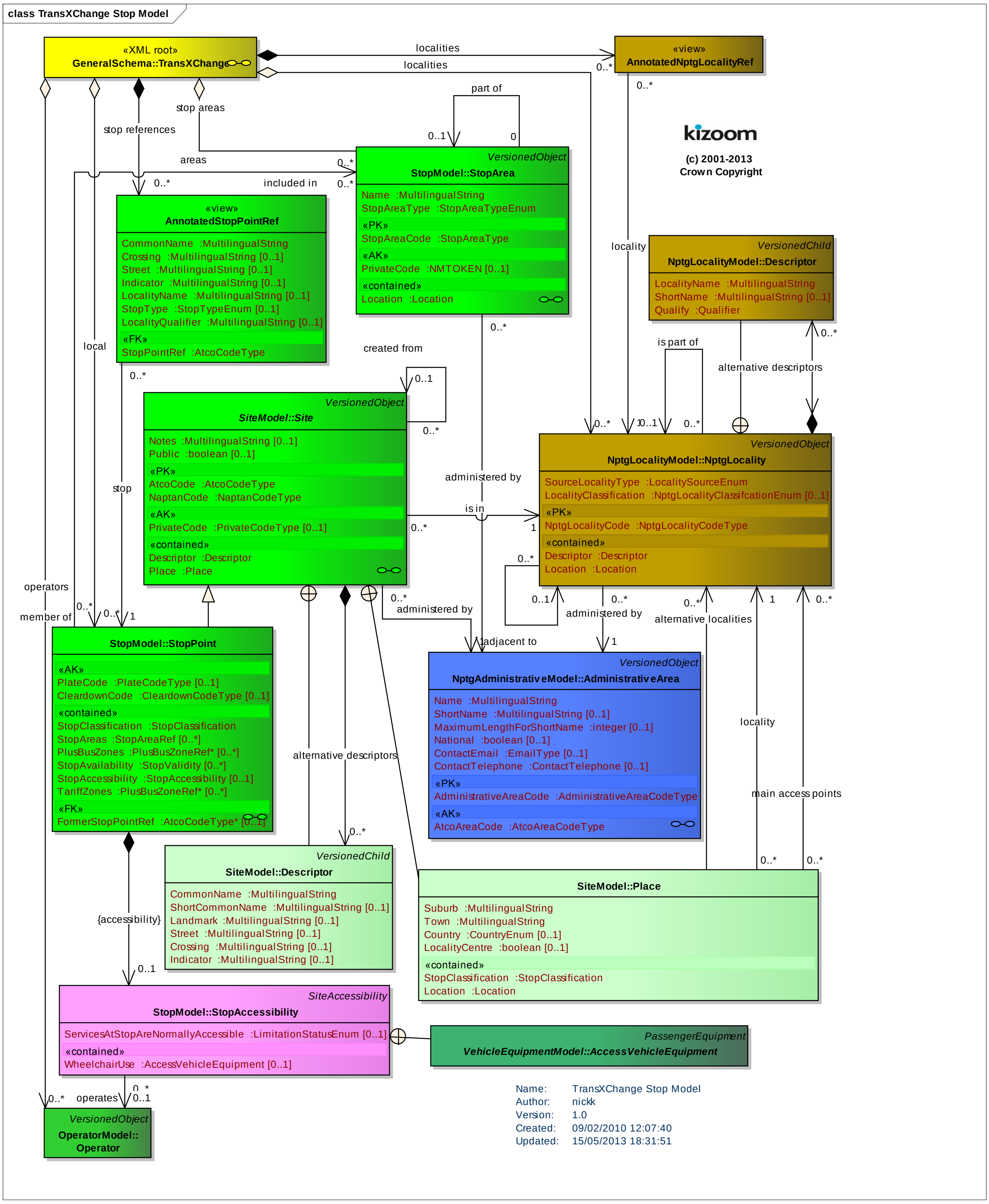


Figure 3-4 – UML Diagram of selected NaPTAN Stop elements

3.1.2 Resolving NaPTAN Stop References

When importing *TransXChange* schedules, an importing application will normally attempt to find the ***StopPoint*** details in the *NaPTAN* database using the NaPTAN identifier, i.e. the ***AtcoCode***, and if found may - depending on the application's purpose - use the database's definition of the stop details in preference to any local definitions. Only if no existing ***StopPoint*** definition is found, will the locally declared definition be used. See *Table 3-1.*

TransXChange Document use of stop	**NaPTAN database**	
	Exists	**Does not exist**
NaPTAN StopPointRef	Resolve to *NaPTAN*	Error
Local *NaPTAN* declaration	Resolve to *NaPTAN*	Use Local definition

Table 3-1 – Resolving Stop References

3.1.3 Variable Stop Allocations

For bus stations where the allocation of stops may vary over time, *TransXChange* supports variable stop allocation. In such cases the journey pattern should reference a *NaPTAN* stop of type BCQ, representing an unspecified stop or bay within the bus station, and then also specify a schedule of allocations to individual bays (i.e. *NaPTAN* stops of type BCT) for a given date, using the ***VariableStopAllocations*** element.

3.1.4 Stop Types

Every NaPTAN ***StopPoint*** has a stop type that indicates its mode and nature, for example, "*on street, bus stop, marked* ". *Figure 3-5* shows, in UML class diagram notation, the stop classification elements of the *NaPTAN* schema. The main items of interest for TransXChange are:

- **OffStreet / BusAndCoach**, for stops in coach stations.
- **OnStreet / Bus** for stops on the street.

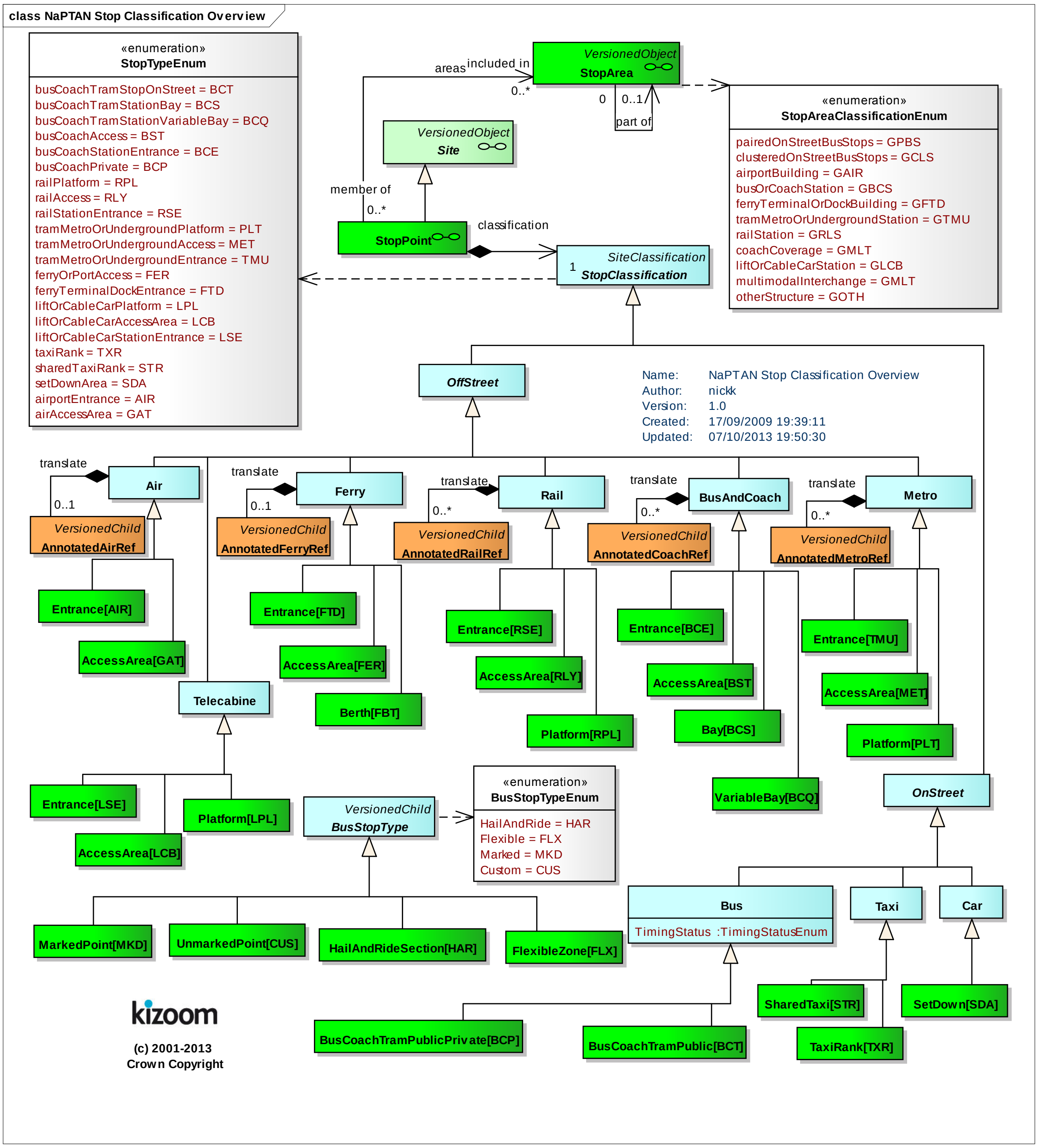


Figure 3-5 – UML Diagram of Stop Classification Model

3.1.5 NaPTAN Bus Stop Types

Figure 3-6 shows, in UML class diagram notation, the ***BusStopType*** elements of the *NaPTAN* schema.

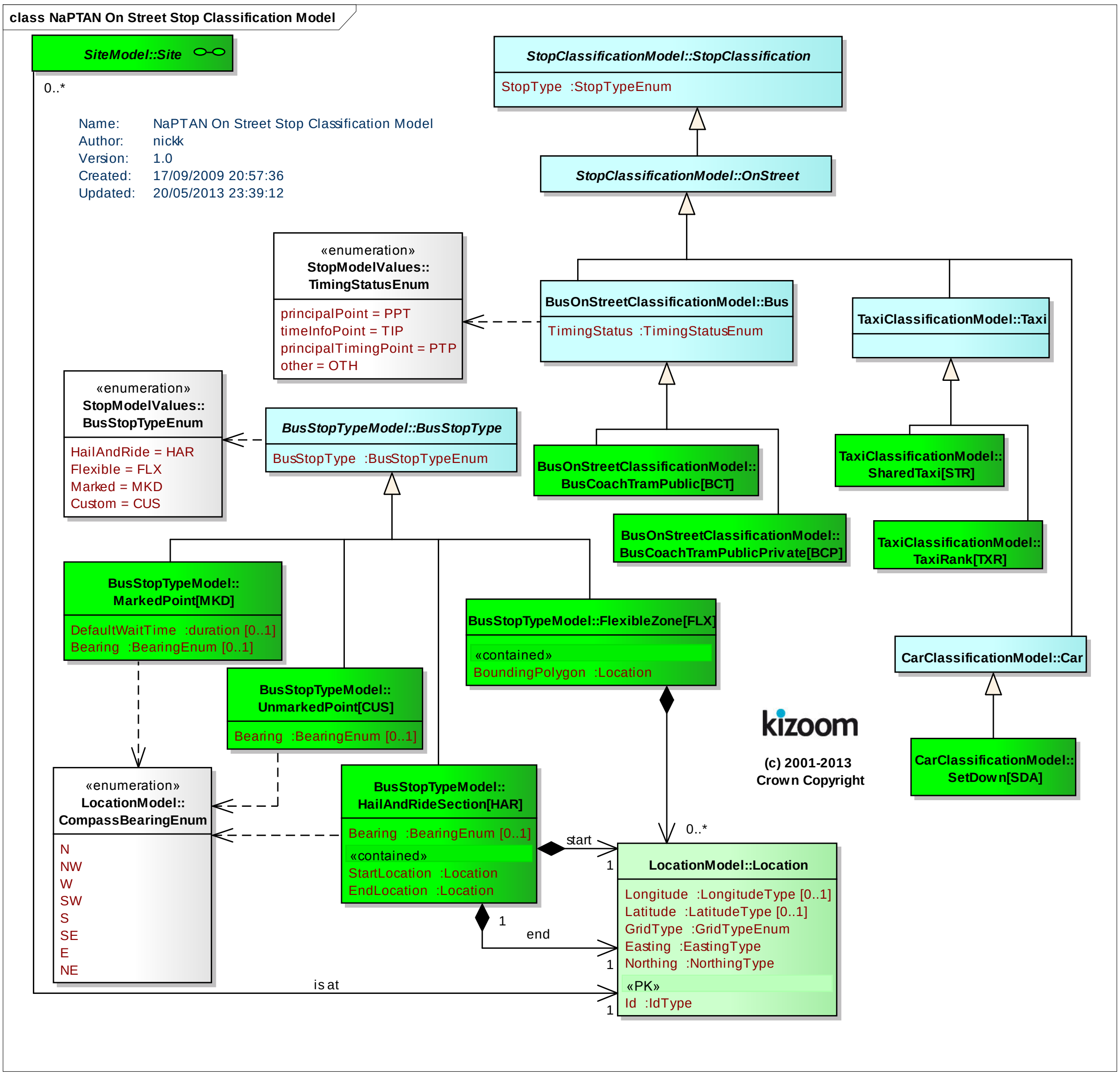


Figure 3-6 – UML Diagram of On-street bus Stop Classification Model

3.1.6 NPTG Administrative Support Types

Figure 3-7 shows, in UML class diagram notation, the base types used by administrative elements of the *NPTG* schema.

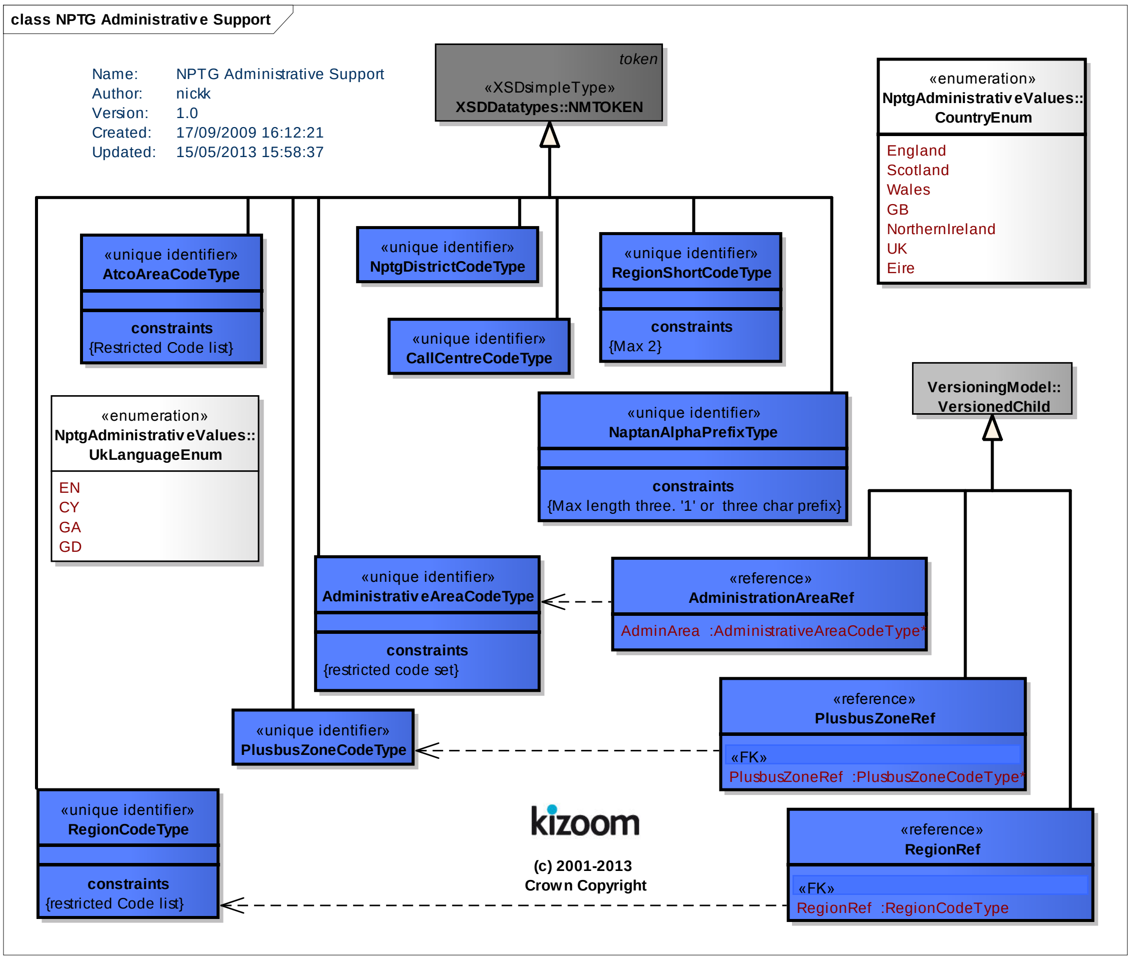


Figure 3-7 – UML Diagram of NPTG Administrative Support types

3.1.7 NPTG Locality Support Types

Figure 3-8 shows, in UML class diagram notation, the Locality types used by elements of the *NPTG* schema.

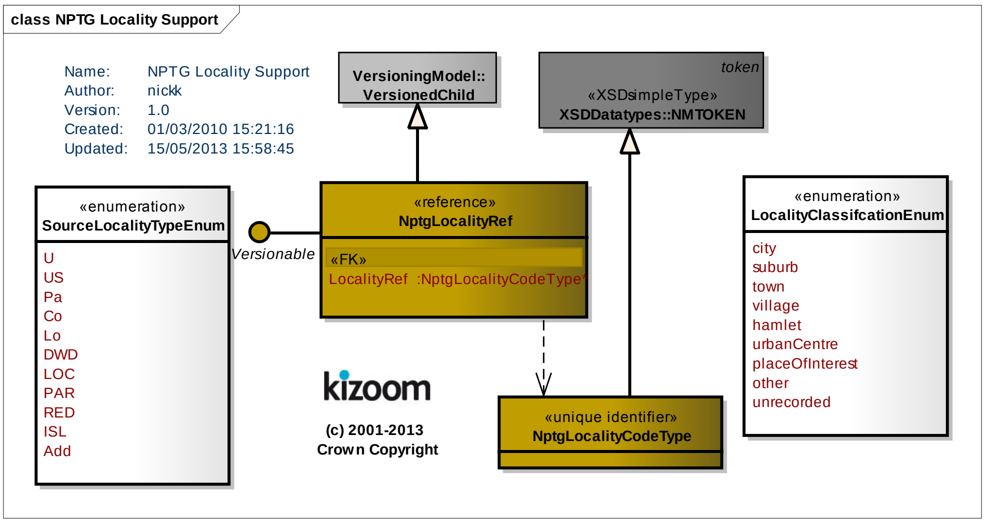


Figure 3-8 – UML Diagram of NPTG Locality Support types

3.1.8 NaPTAN Support Types

Figure 3-8shows, in UML class diagram notation, the base types used by elements of the *NaPTAN* schema.

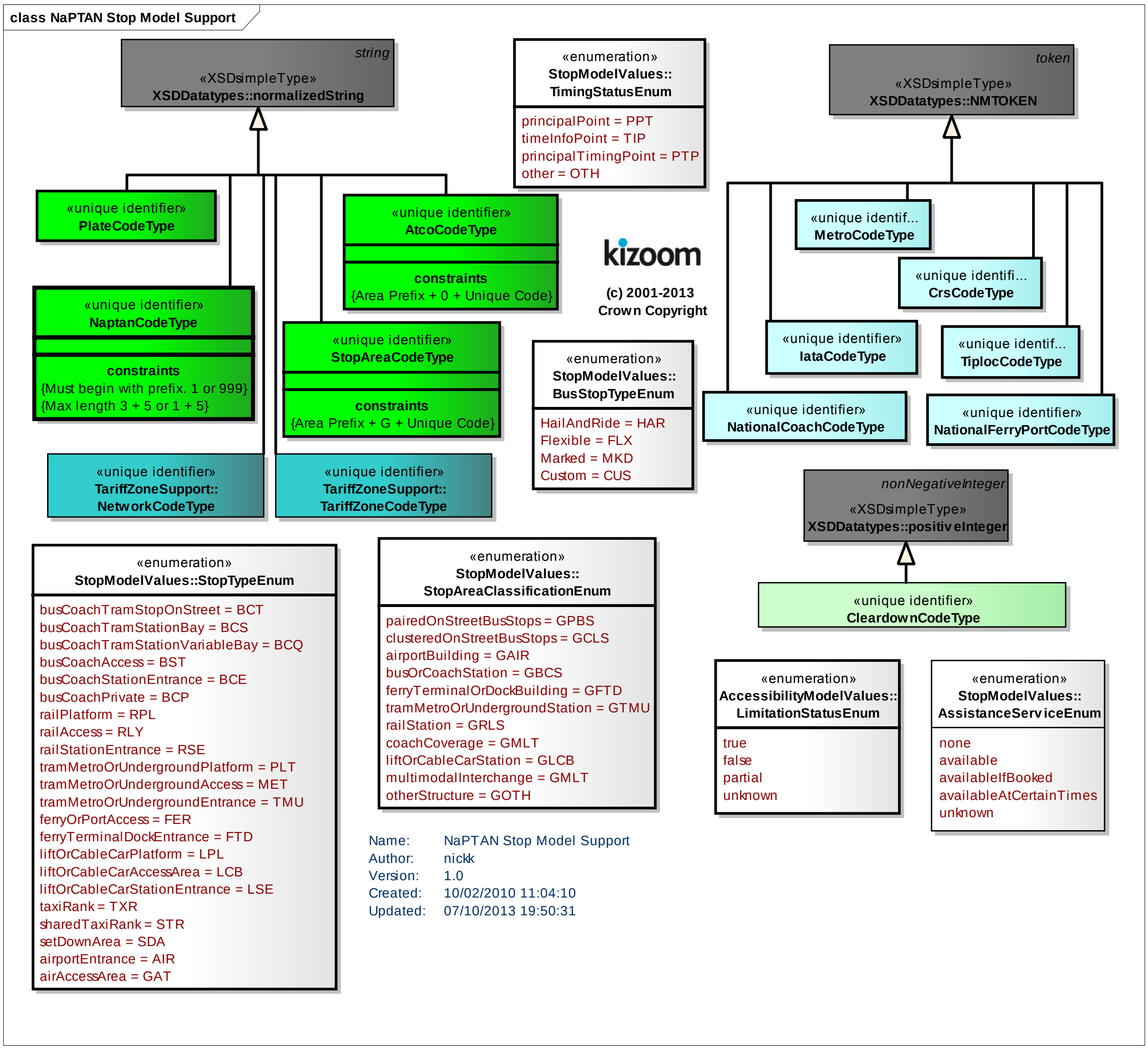


Figure 3-9 – UML Diagram of NaPTAN Support types

3.1.9 NaPTAN Location Support Types

Figure 3-10shows, in UML class diagram notation, the location data types used by the NPTG, *NaPTAN* & TXC schema.

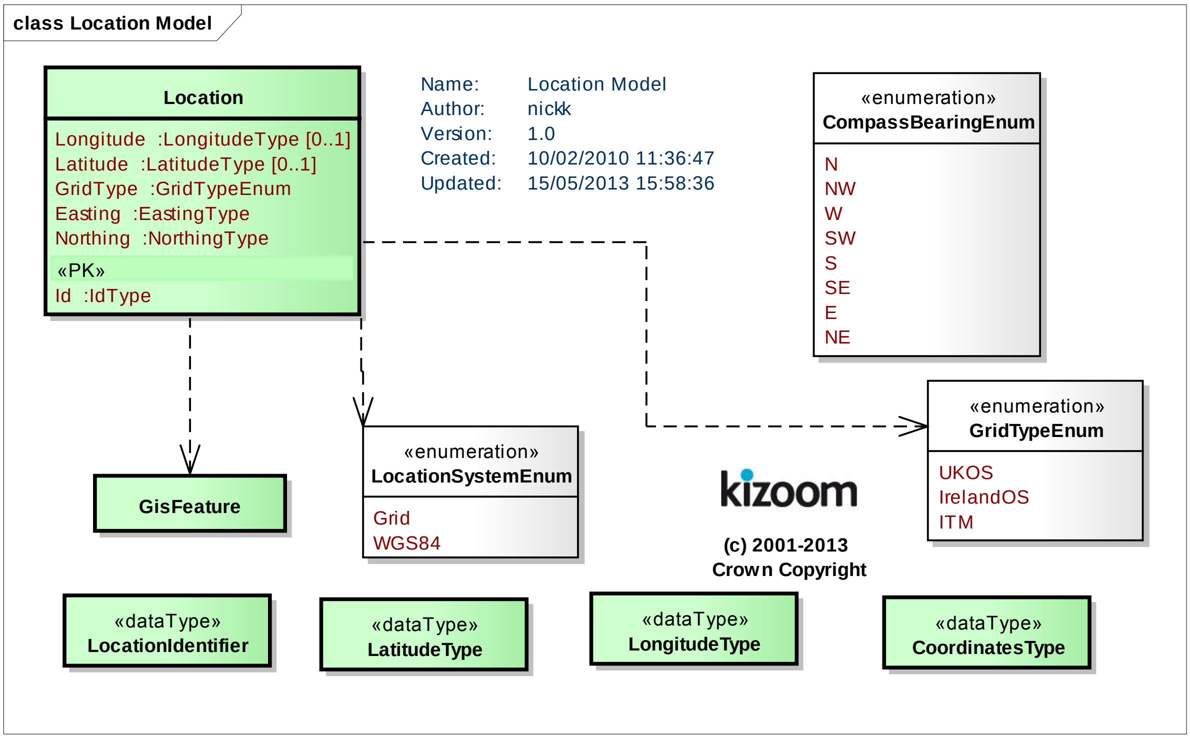


Figure 3-10 – UML Diagram of NaPTAN Location Support Types

3.2 The Route and Service Supply Model

TransXChange describes a bus service using a model made up of three distinct layers or 'levels of discourse' (see *Figure 3-11* for an UML diagram):

1. A ***Route***; described as a sequence of route links connecting individual stops. For *TransXChange,* all stops are defined as being *NaPTAN* points, so a route describes a path in '*NaPTAN* space'; a distinct frame of reference made up of Public Transport Access Nodes (PTANs), which is semantically distinct from any given coordinate system, but which can be projected onto geospatial coordinate systems and mapping layers using ***Track*** elements.
   - The ***RouteLink*** instances are grouped using a ***RouteSection***, allowing the reuse of whole sequences of links in different routes.
   - ***Track*** elements record both the plot of the route at non-*NaPTAN* points, and associations with mapping layer identifiers, such as OS TOIDS.

2. A ***JourneyPattern***: a path over the route made up of a number of journey pattern timing links, each with timing information (and other optional operational data) ascribed to them. All timing information is relative (for example, *'+5 minutes'*).
   - Each end of a ***JourneyPatternTimingLink*** can have stop usage information associated with it on a ***JourneyPatternStopUsage*** element, specifying the activity at stop, and other service information.
   - The timing links are grouped using a ***JourneyPatternSection***, allowing the reuse of whole sequences of links in different patterns.
   - The links of a ***JourneyPattern*** must traverse the same stops in the same sequence as the links of any ***Route*** associated with the ***JourneyPattern***. However a ***JourneyPattern*** need not cover the whole ***Route***; it may project onto just a contiguous subset of the links of the route, omitting route links at either or both ends.

3. A ***VehicleJourney***: a traversal of a specific journey pattern at a specific time: again modelled as a sequence of timing links connecting *NaPTAN* stops, using ***VehicleJourneyTimingLink*** and ***VehicleJourneyStopUsage*** elements.
   - Each vehicle journey has an absolute start time (e.g. *'13:02'*) specified: this can be combined with the timing information from each timing link to derive the actual passing times of departure and arrival at each timing point.
   - The public identifier of a ***VehicleJourney*** is given by a ***Line***. One or more ***Line*** instances may be associated with a service, and a ***VehicleJourney*** must reference one of its service's lines.
   - The link sequence of a ***VehicleJourney*** must exactly correspond to the link sequence of the underlying ***JourneyPattern***; that is, each ***VehicleJourneyTimingLink*** must project onto a corresponding ***JourneyPatternTimingLink***.

The *Transmodel* principles underlying the *TransXChange* Route and Service Supply model are summarised in Section 13.1, and divergences from Transmodel usage are listed.

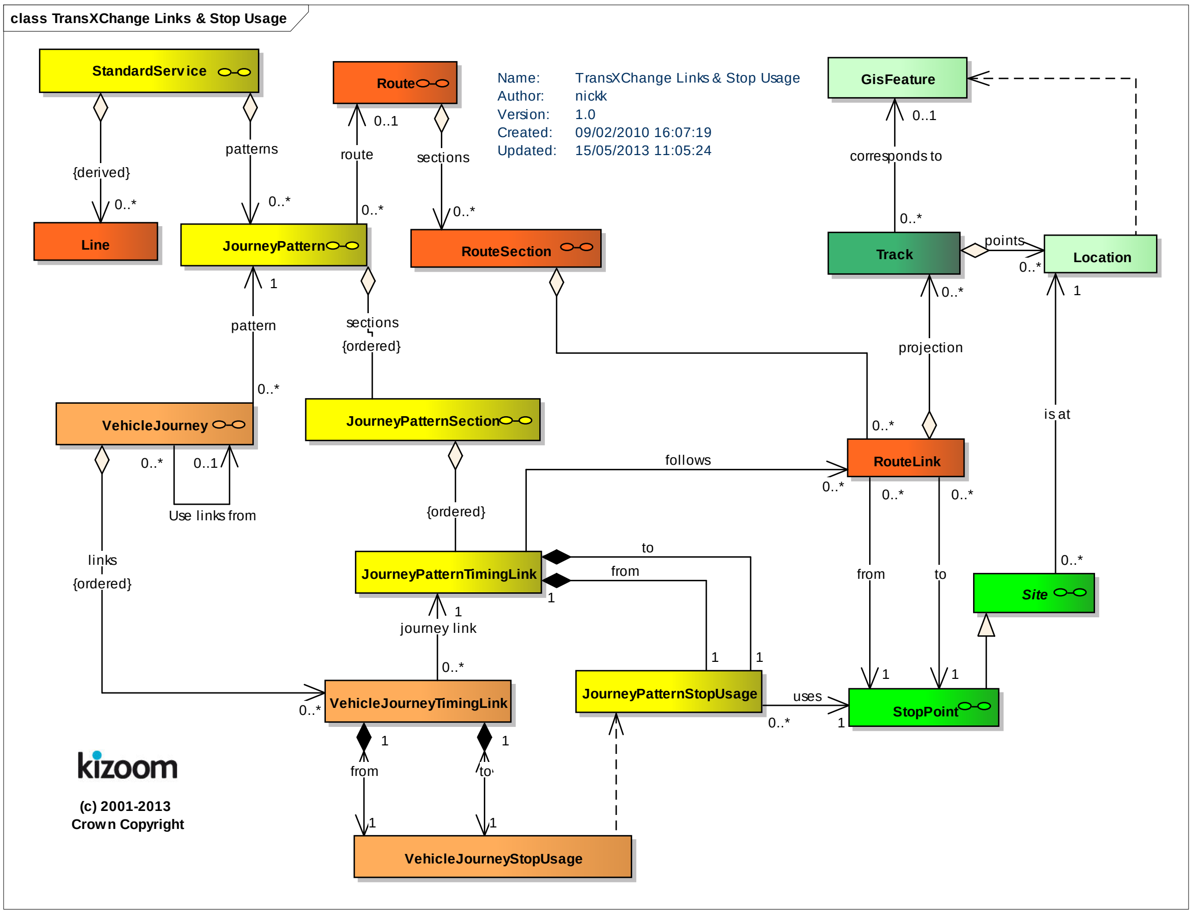


Figure 3-11 – UML Diagram of Route, JourneyPattern and VehicleJourney Models

3.2.1 Model Layer Concerns

Figure 3-12 illustrates how the each layer has a separate concern of the model:

1. The ***Route*** describes the stops, stop sequence, and the physical track between them.
2. The ***JourneyPattern*** adds in timing information; how long each link takes to run, how long to wait at each stop, and the allowed activities at each stop.
3. A ***VehicleJourney*** specifies a start time: this is used to compute actual passing times for each stop in the journey pattern, taking into account the run and waiting times. The vehicle journey can override the run time, wait time and activity from the journey pattern values for its own journey, but not change the stop sequence.
4.

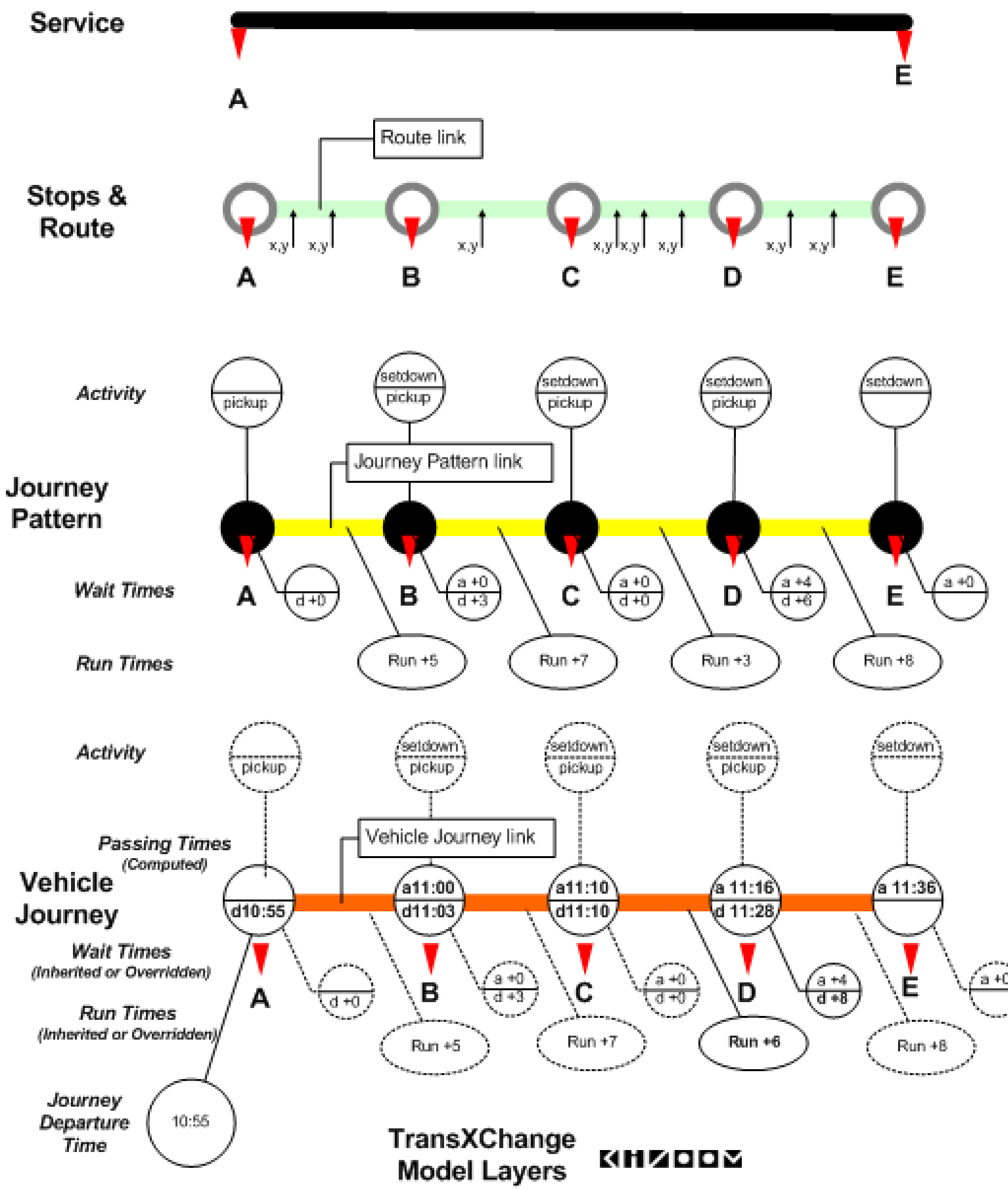


Figure 3-12 – Service Model Layer Concerns

3.2.2 Summary of Route & Supply Model Elements

Each of the three layers is made up of three sets of broadly equivalent elements:

(i) Ordered collections, i.e. sequences, of links (*Patterns* and *Sections*).
(ii) Links (Route *Links* and Timing *Links*).
(iii) Link ends (*Stop Usages*).

Table 3-2 summarises the route and supply model elements, showing the simple one-to-one correspondences between equivalent elements in the different layers. The simple correspondence makes it straightforward to project between the route, journey pattern and vehicle journey layers. There are explicit references between elements in the pattern and link columns, which can be used to derive an implicit projection of the section and stop usage.

Ordered Link Sequence		**Link**	**Link end**
Pattern	(Section)	Link	Stop Usage
Route	***RouteSection***	***RouteLink***	***StopReference***
(AbstractJourneyPattern)	**-**	***AbstractTimingLink***	***AbstractStopUsage***
JourneyPattern	***JourneyPatternSection***	***JourneyPatternTimingLink***	***JourneyPatternStopUsage***
VehicleJourney	**-**	***VehicleJourneyTimingLink***	***VehicleJourneyStopUsage***

Table 3-2 – Correspondence between Links and Nodes

3.2.3 Projection between Levels of Discourse

Figure 3-13 shows a schematic example of links at different levels of discourse and the correspondences between them.

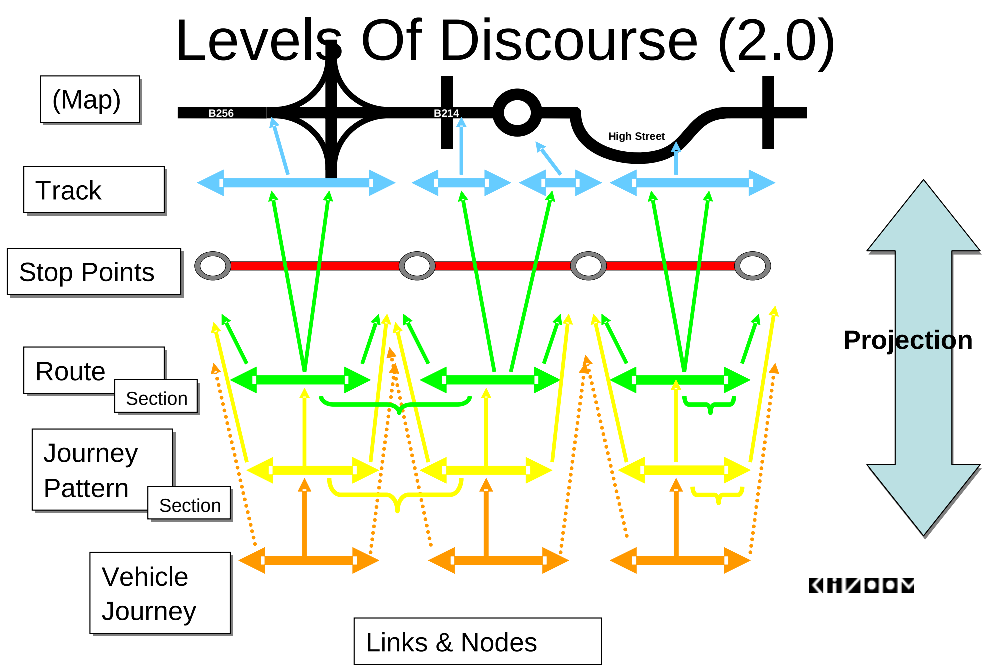


Figure 3-13 – Correspondence between Links at Different Levels

3.3 Route Model

3.3.1 Introduction to the Route Model

Figure 3-14 introduces the elements used to represent a ***Route***. These comprise ***Routes***, ***RouteSections*** (Reusable sequences of ***RouteLinks***), ***and RouteLinks***. The projection of the route is described by a ***Track*** made up of ***Instruction*** and ***Feature*** elements.

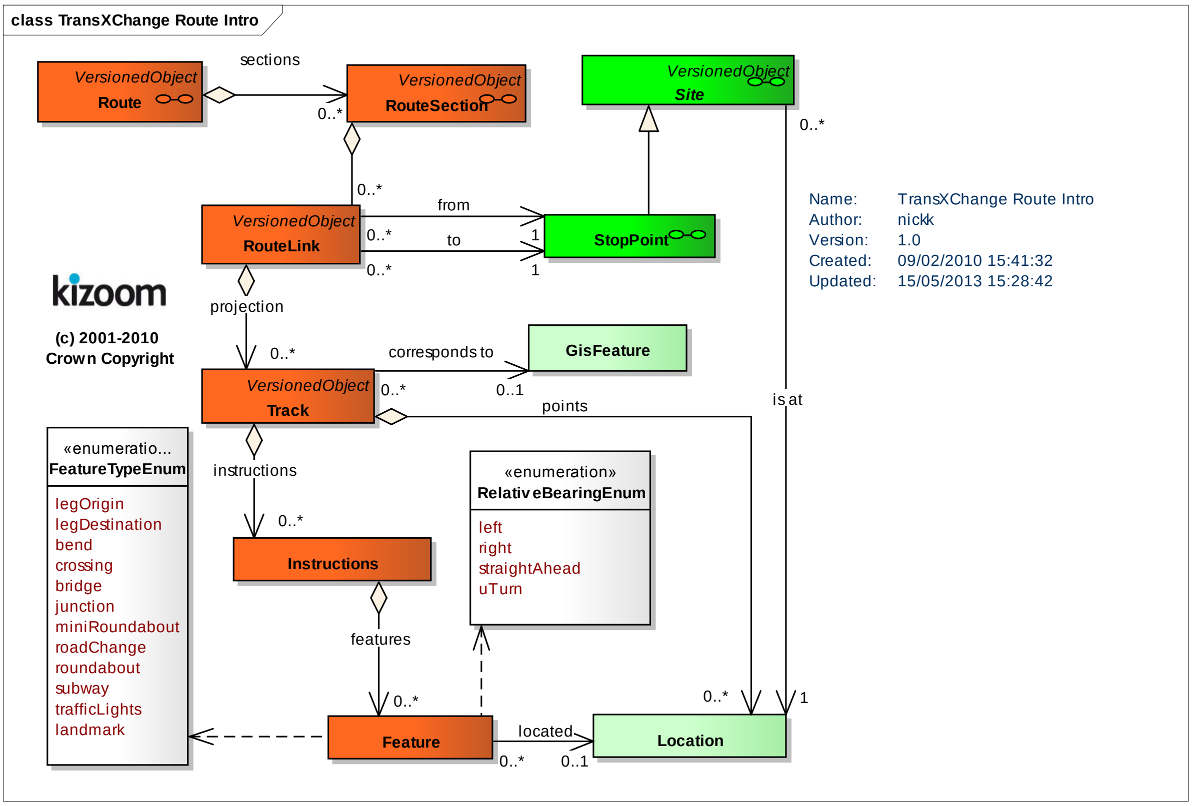


Figure 3-14 – UML Diagram of Route Model: Introduction

3.3.2 Route Model Details

Figure 3-15 elaborates *Figure 3-14* to show further attributes.

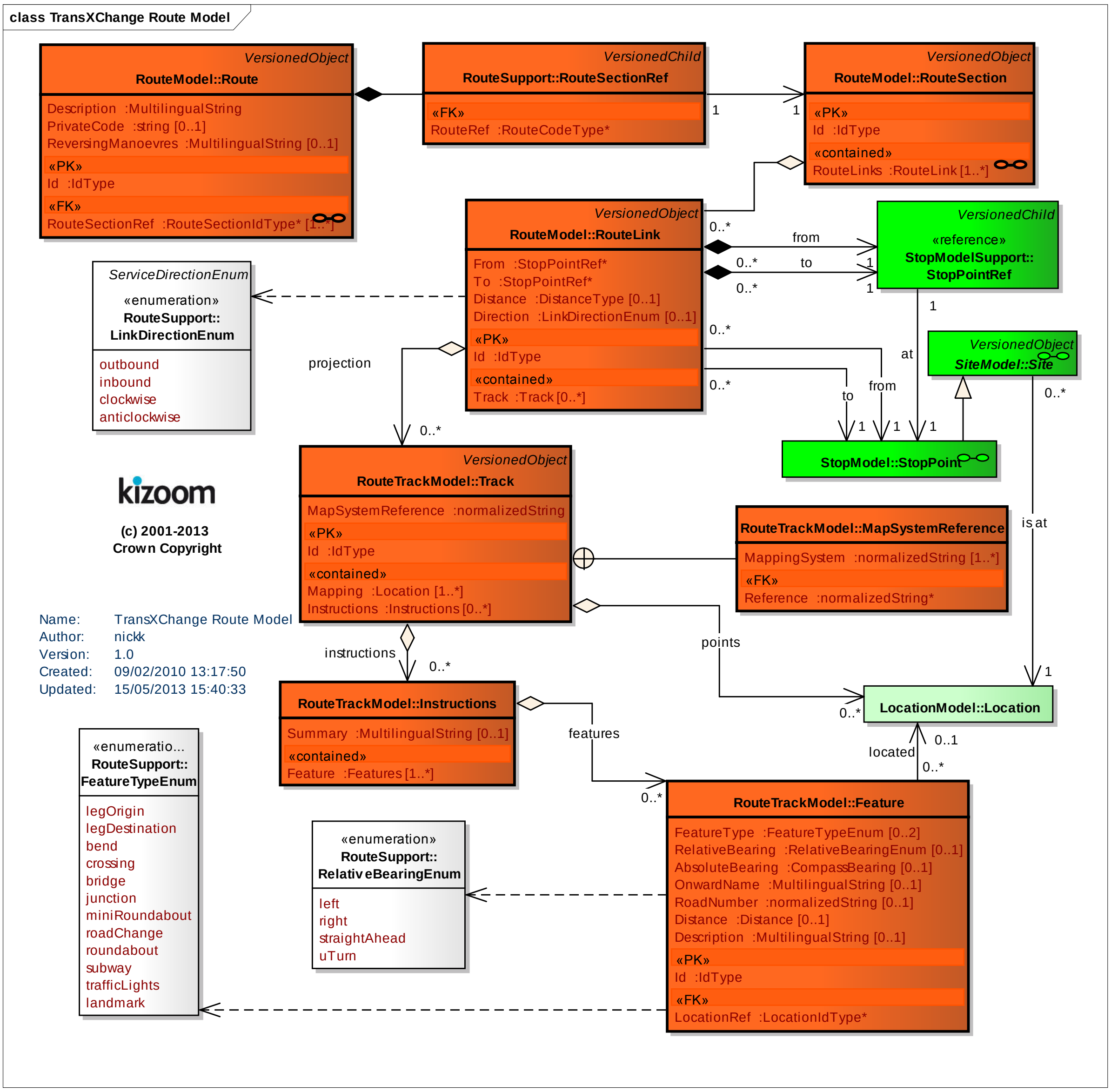


Figure 3-15 – UML Diagram of Route Model

3.3.3 Tracks

The *TransXChange* Track model describes details about the physical course of a ***RouteLink,*** in particular the collection of spatial points needed to plot the route unambiguously in sequence on a map of the road network, for example using a 'snap to track' algorithm. As well as such a ***Mapping***, a ***Track*** can also be associated with a reference to an external mapping system using a ***MapSystem-Reference*** element, allowing the projection of links onto geospatial map layers. ***Track*** features can also be used to describe any manoeuvre involved in navigating a route link, such as a U-turn.
The Track model allows a rich description of a route to be provided; it is intended for general purpose data exchange. For a Registration a level of Track detail should be given sufficient to unambiguously plot the route on a map using OSGR data – using both points and/or TOIDS.

It is a requirement of registration that adequate spatial data is provided as to plot routes on an OS map in a useful way: there should be intermediate coordinates for a reasonably high level of resolution.

Figure 3-14 shows a UML structure diagram of the elements used to describe tracks. Tracks can contain two different types of description:

- A ***Mapping*** describes the geospatial plot of the route link as two or more ***Location*** elements that provide point coordinates for the track between *NaPTAN* stop points.
- An ***Instructions*** instance provides an optional additional structured description of the steps involved in traversing the track as a sequence of ***Feature*** instances. For example '*Turn left at roundabout into Mary Street*'.

3.3.4 Track Example

As a simple example, consider a ***RouteLink*** that runs along the B205 and B257, represented by a two ***Track*** instances.

- Each ***Track*** instance has a ***Mapping*** instance that describes the course of the track. tr1 has two points (g_1, g_2)) and tr2 has seven points (g_3 to g_7) respectively; each point is a ***Location*** instance that describes a point of the track.

- Each ***Track*** has an ***Instructions*** instance containing an ordered collection of ***Feature*** instances.

- Each ***Feature*** instance describes a step needed to traverse the track, and references a ***Location*** instance from the Table 3-3 shows a sample of the ***Feature*** instances.

Track	Location Ref	Feature Type	Relative Bearing	Absolute Bearing	Onward Name	Road Number	Dist-ance	Description
Tr1	*G_1*	*legOrigin*	*straightAhead*	*N*	*Victoria Road*	*B205*	*300m*	*Proceed 300m North down Victoria road (B205.)*
	G_2	*junction*	*left*	*W*	*Albert Road*	*B205*	*500m*	*Turn left into Albert road (B257) and head west 500m.*
Tr2	*G_3*	*landmark*	*straightAhead*	--	--		--	*Hospital on left*
	G_4	*bend*	*right*	*NW*	*Albert Road*	*B257*	--	*Follow bend to right in Albert Road*
	G_5	*roadChange*	*straightAhead*	*NW*	*George Road*	*B257*	*400m*	*Continue 400m down George Road (B257)*
	G_6	*roundabout*	*left*	*SW*	*Mary Street*	*B257*	--	*Turn left at roundabout into Mary Street*
	G_7	*crossing*	*straightAhead*	--	*Bill Alley*	*B257*	--	*Cross over Bill Alley*
	G_8	*bridge*	*straightAhead*	--	*Mary Street*	*B257*	--	*Pass under bridge*

	G_9	*legDestination*	*straightAhead*	*S*	*Mary Street*	*B257*	*600m*	*Continue straight ahead 600m South down Mary Street*

Table 3-3 – Example Track Instructions

3.3.5 Route Model Hierarchy

Figure 3-16 shows the inheritance hierarchy for the Route model elements.

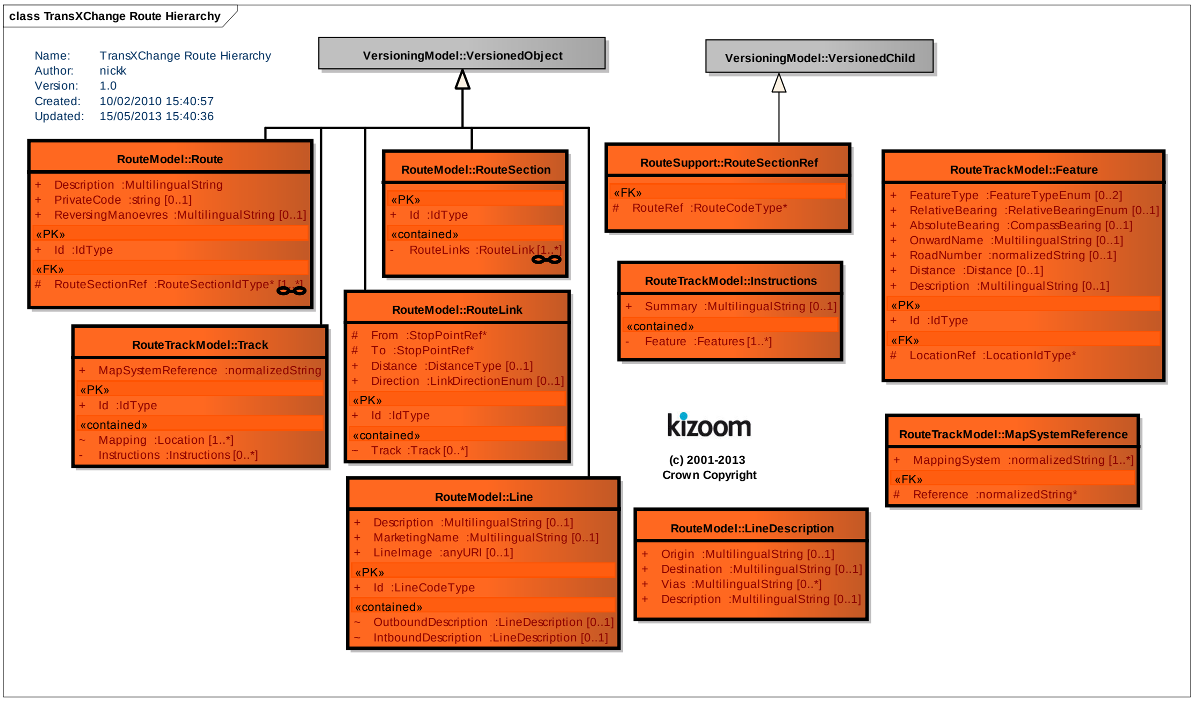


Figure 3-16 – UML Diagram of Route Model Element Hierarchy

3.4 Journey Pattern Model

3.4.1 Introduction to the Journey Pattern Model

Figure 3-24 introduces the elements used to represent a ***JourneyPattern***. These comprise ***Journey-Patterns***, ***JourneyPatternSections*** (Reusable ordered sequences of ***JourneyPatternTiming-Links***), and ***JourneyPatternTimingLinks***. Each ***JourneyPatternTimingLink*** connects two stops: attributes for each link end may be specified by a ***JourneyPatternStopUsage***.

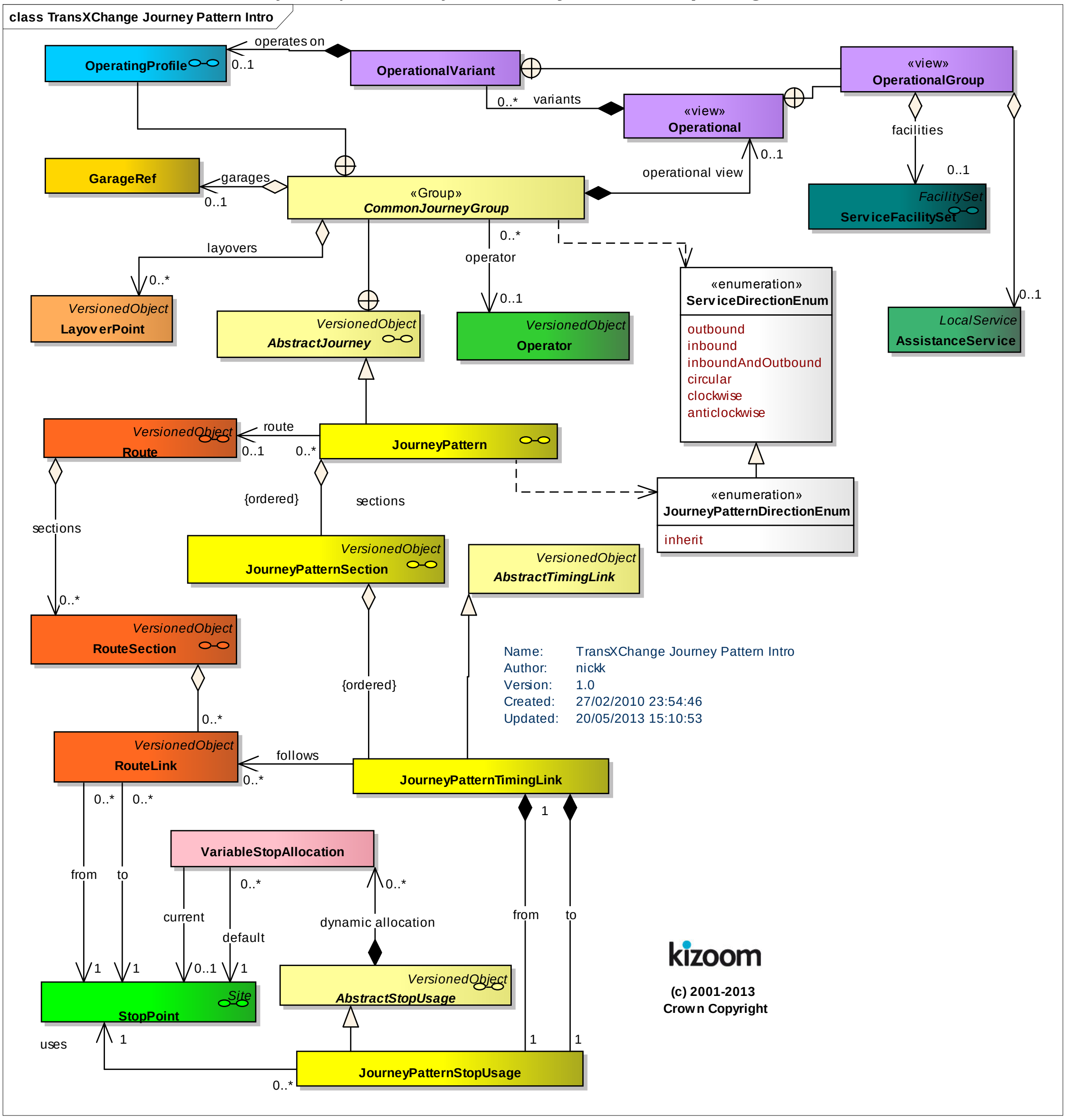


Figure 3-17 – UML Diagram of Journey Pattern Model: Introduction

3.4.2 Journey Pattern Model Details

Figure 3-18 elaborates *Figure 3-24* to show the detailed attributes of ***JourneyPattern*** elements.

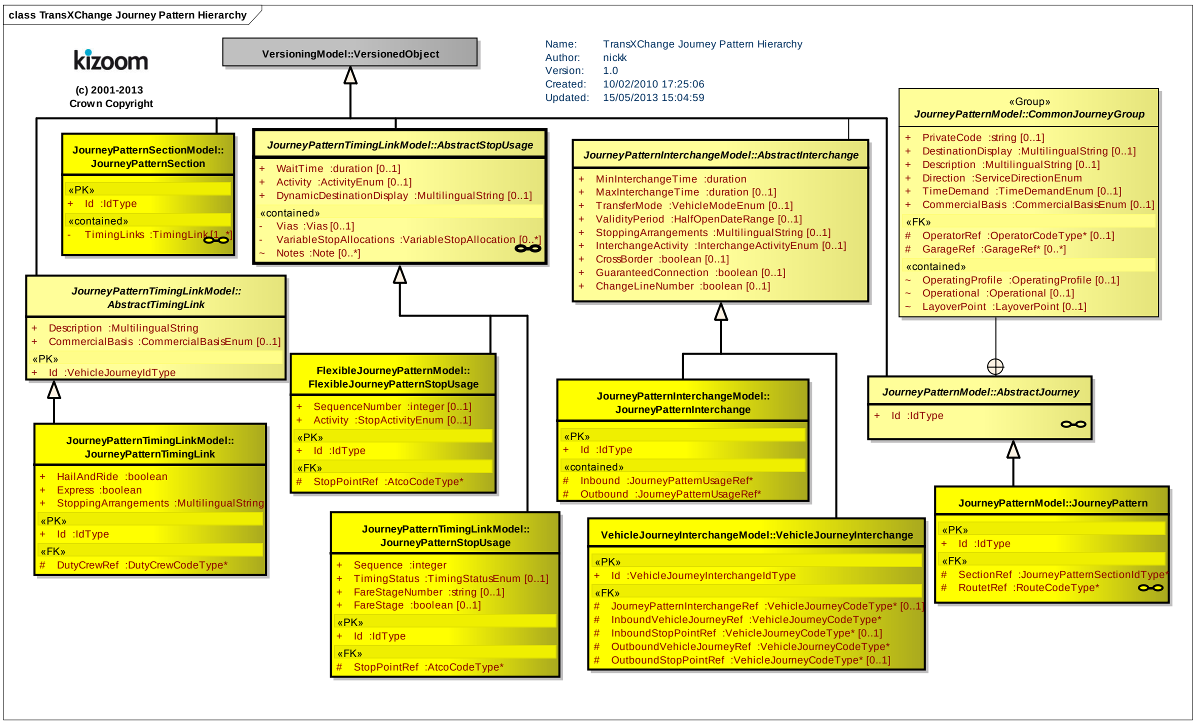


Figure 3-18 – UML Diagram of Journey Pattern Model: Details

3.4.3 Journey Pattern Model Hierarchy

Figure 3-19 shows the inheritance hierarchy for the ***JourneyPattern*** model elements.

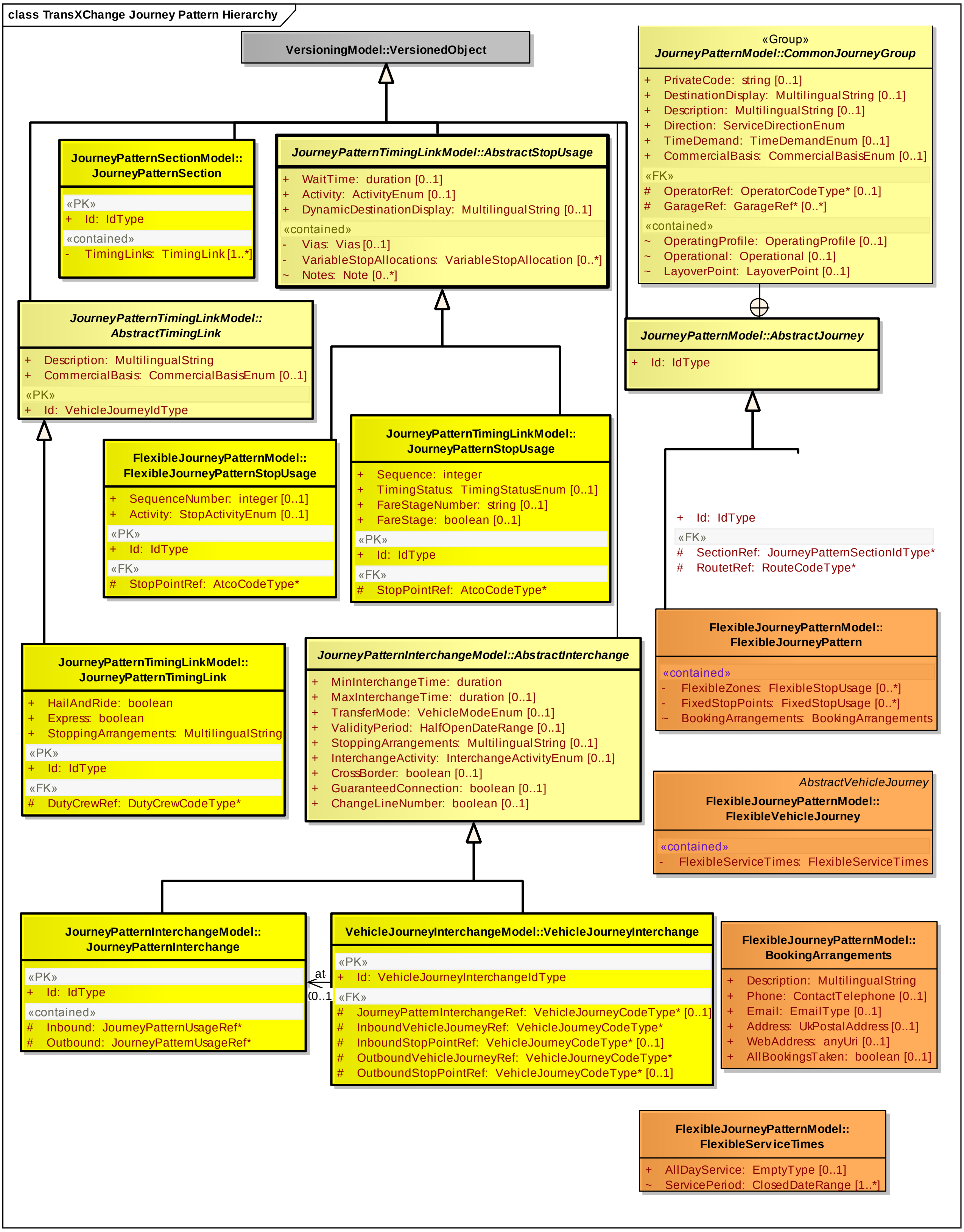


Figure 3-19 – UML Diagram of Journey Pattern Model Element Hierarchy

3.5 Vehicle Journey Model

3.5.1 Introduction to the Vehicle Journey Model

Figure 3-24 introduces the elements used to represent a ***VehicleJourney***. These comprise ***AbstractVehicleJourney***, ***VehicleJourney, ConnectingVehicleJourneys*** (Lightweight ***Vehicle-Journeys*** for describing external interchanges) and ***VehicleJourneyTimingLinks***).

Each ***VehicleJourneyTimingLink*** connects two stops: attributes for each link end may be specified by a ***VehicleJourneyStopUsage***. A ***Frequency*** element further describes the intervals of Frequency Based ***VehicleJourneys. A DeadRun (***see separate section later***)*** describes a vehicle positioning run that does not appear in the public timetable***.***

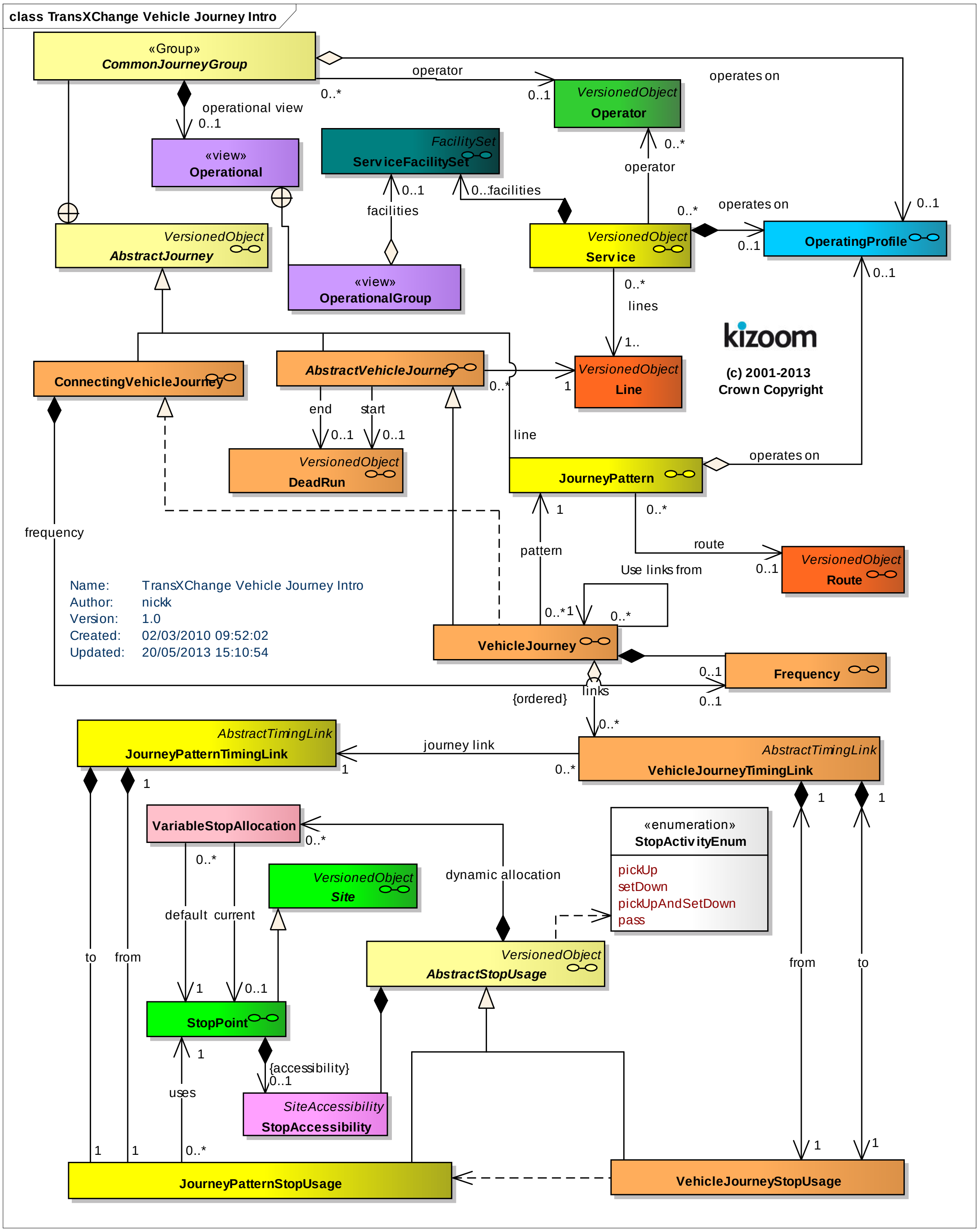


Figure 3-20 – UML Diagram of Journey Pattern Model: Introduction

3.5.2 Vehicle Journey Model Details

Figure 3-15 elaborates *Figure 3-14* to show the attributes of ***VehicleJourney*** elements.

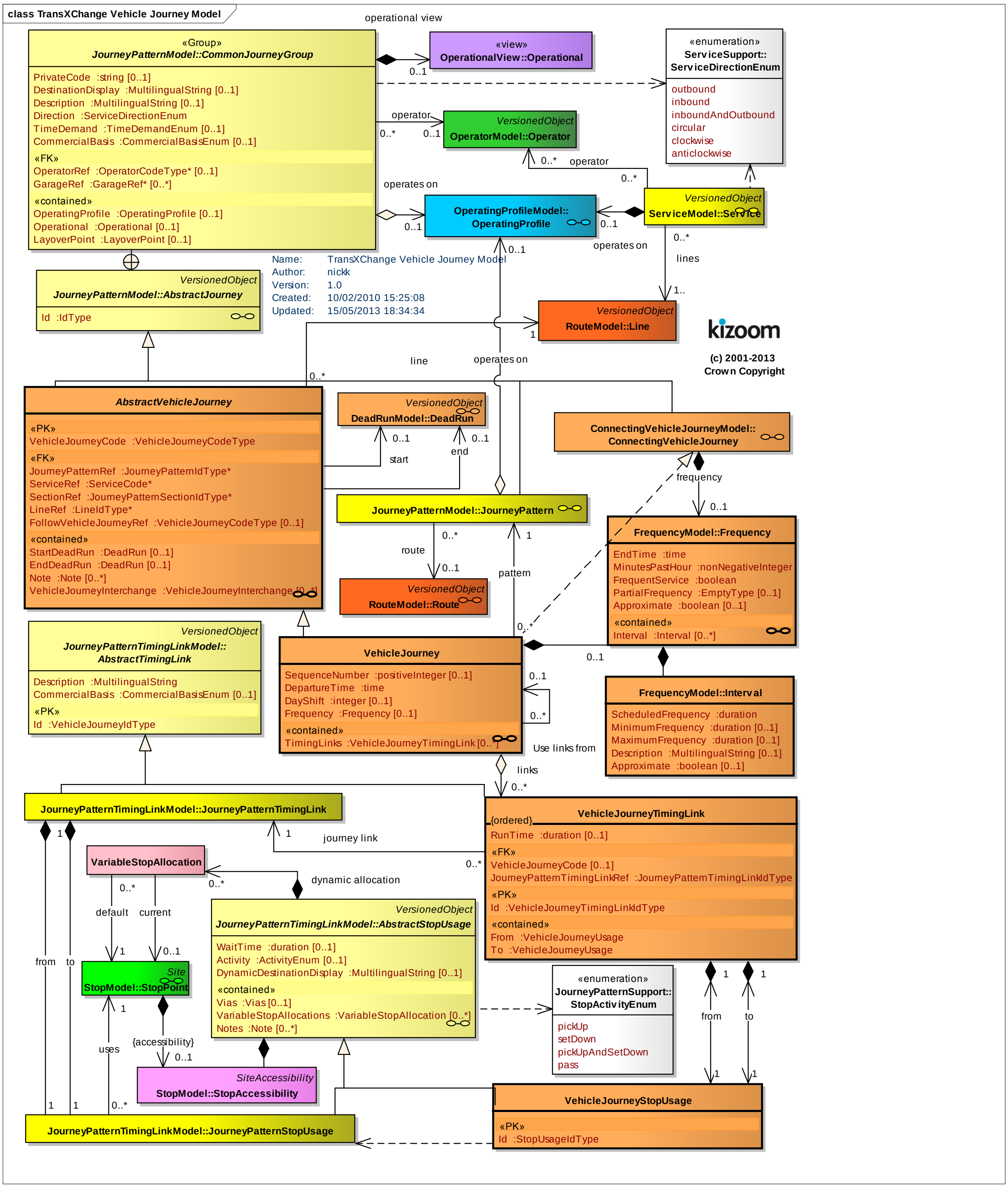


Figure 3-21 – UML Diagram of Vehicle Journey Model: Details

3.5.3 Connecting Vehicle Journey Model

Figure 3-16 shows the model for a ConnectingVehicleJourney. A connecting vehicle allows details of a connecting journey to be referenced.

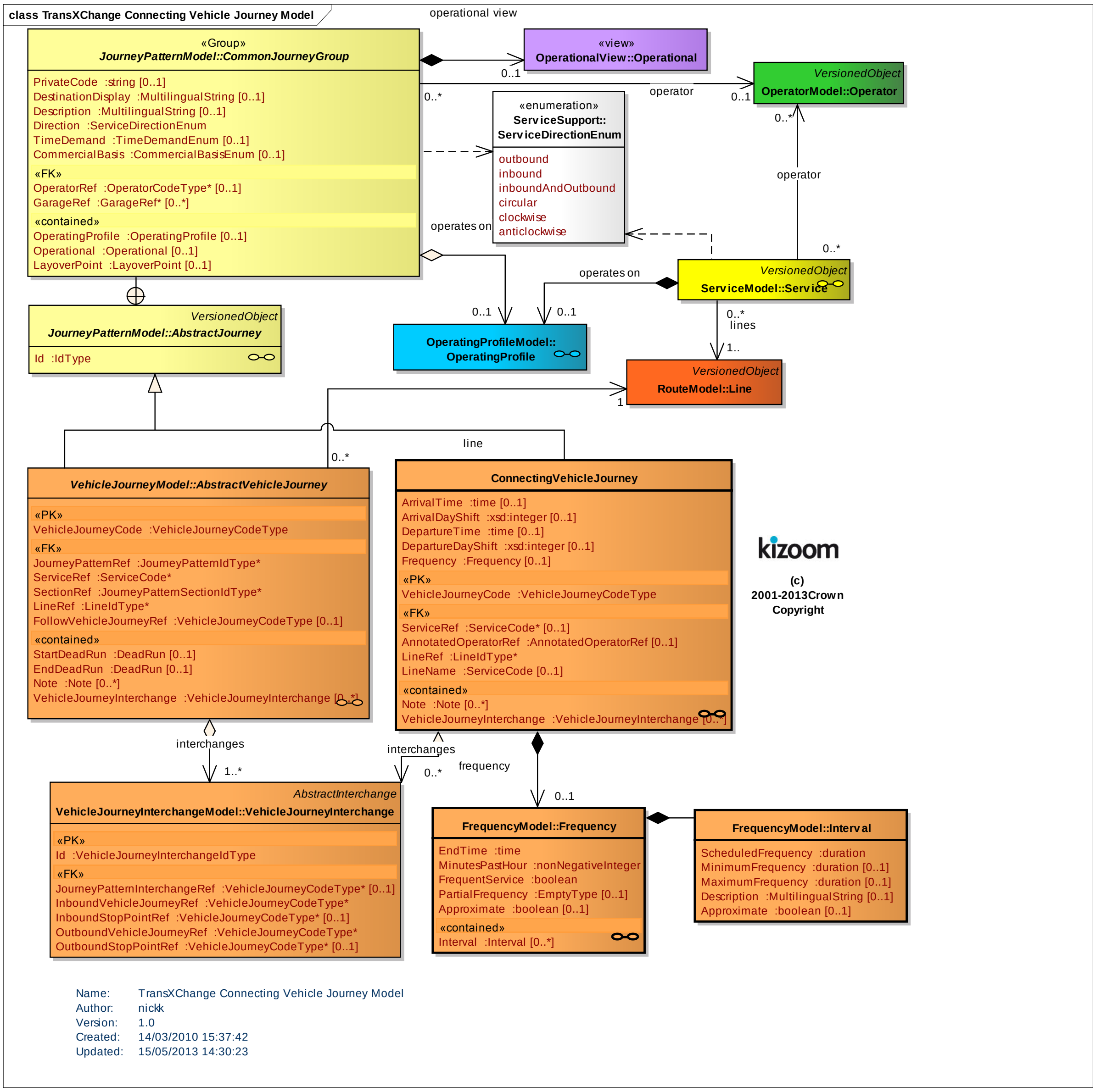


Figure 3-22 – UML Diagram of Connecting Vehicle Journey Model

3.5.4 Vehicle Journey Model Hierarchy

Figure 3-16 shows the inheritance hierarchy for the ***VehicleJourney*** model elements.

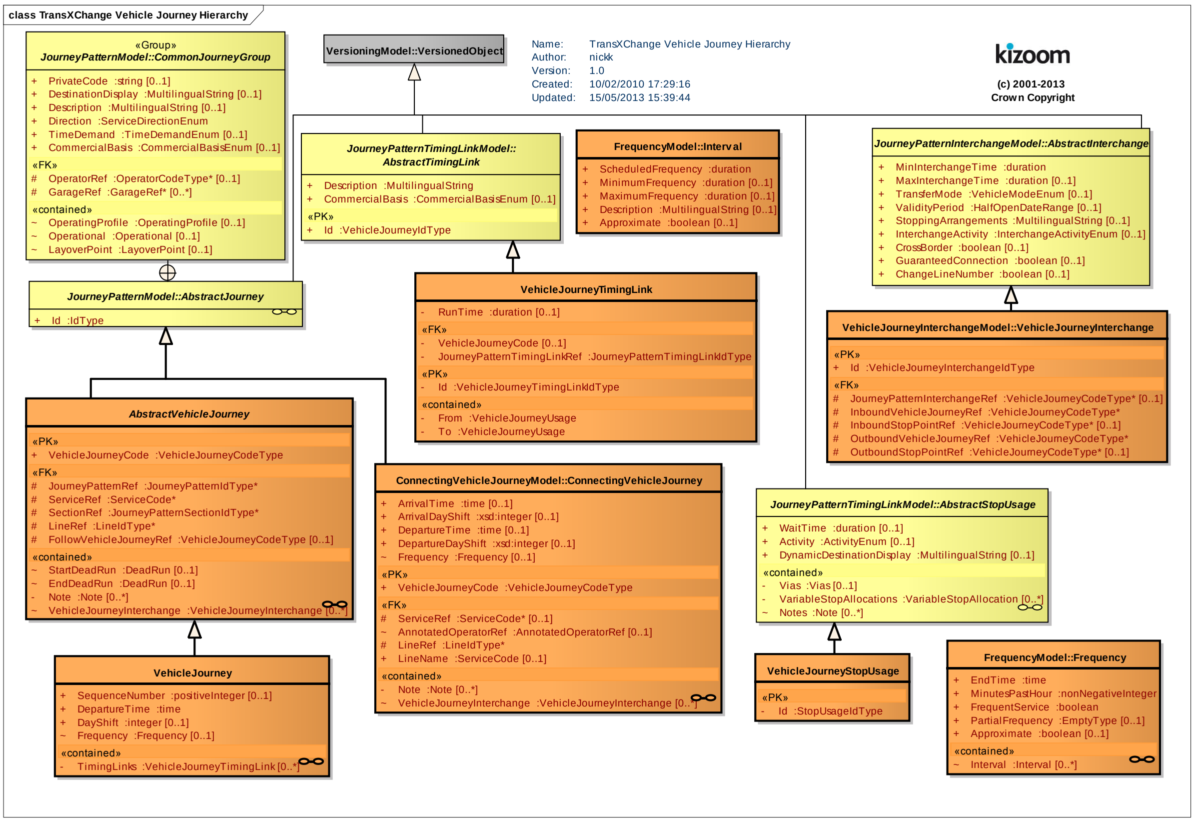


Figure 3-23 – UML Diagram of Vehicle Journey Model Element Hierarchy

3.6 The Use of Links in TransXChange

In *Transmodel*, a journey can be regarded either as an ordered list of stops, or as an ordered list of links between the stops: both views can be derived from the underlying *TransXChange* representation of a journey pattern and vehicle journey as a list of timed links. In *TransXChange,* a 'timing link in link sequence' representation is used (see discussion of *Transmodel* terminology and concepts in section 13.2), as this holds more information than a simple stop list, and can be projected exactly onto a spatial route; it can readily be transformed by applications into a list of stops and passing times if needed.

The following *Transmodel* principles apply to the use of journey patterns in *TransXChange:*

1. *There should be a separate journey pattern for each physical route followed*, i.e. a sequence of timing links between stops defining a unique sequence of stops.

2. *A vehicle journey must always follow a journey pattern*.

3. *A vehicle journey must visit all the stops of a journey pattern*, with two qualifications (which are not strictly *Transmodel* - see 13.2*)*:
    a. **Short working** of the underlying journey pattern is allowed, i.e. truncation of one or more stops at either or both ends.
    b. **Express journeys** over a service pattern are allowed – i.e. provided a journey traverses a link and goes past a stop, it may specify an activity of '*pass'* to omit a particular stop.

The following further principles apply to the use of links to represent journey patterns in *TransXChange:*

4. A vehicle journey *need specify explicitly only those timing links that are different from the underlying journey pattern.* Other vehicle journey links may be *implicit*, that is derived automatically from the underlying journey pattern. In many cases, no *explicit* concrete links need be specified in a vehicle journey.

5. *A vehicle journey may reference all the links of another vehicle journey. In this case all the link usage must be implicit,* that is, all of the links of the referenced journey are used with the same values as in the referenced journey*.* If the vehicle journey needs to make modifications to links or link properties, it should be based directly on an underlying journey pattern, and not reference another vehicle journey for some links and make further changes.

6. Timing links may have a number of different '*successive'* properties that change over successive steps of the journey pattern, for example, destination headings, duty crews, and fare stages. The properties may be set on individual links at both the journey pattern and vehicle journey level. *Once a successive property (*such as a dynamic destination heading*) is set on a specific link (*or individual link end*), it is considered to be in effect on successor links in the journey until any different value is encountered on a subsequent link.* Link values on successor vehicle journey links may either be set explicitly, or be inherited from a parent journey pattern link.

3.6.1 Structure Example of a Schedule with one Pattern and Two Journeys,

Figure 3-24 shows a simple route, with five stops connected by four links.

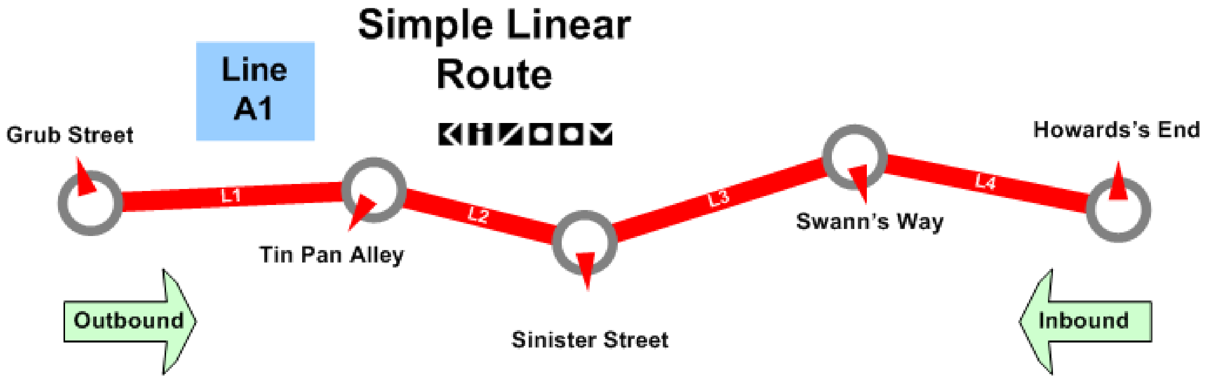


Figure 3-24 – Simple Route Map

Table 3-4 shows an example timetable of a service running over the route, with two vehicle journeys running between each of the five stops.

Name\Line		**A1**	**A1**
Grub Street	d	8:02	10:02
Tin Pan Alley	d	8:12	10:12
Sinister Street	d	8:37	10:37
Swans Way	d	8:45	10:45
Howard's End	a	8:55	10:55

Table 3-4 – Structure Example of a Schedule

Table 3-5 shows this same timetable annotated with the XML element instances needed to represent it in a *TransXChange* XML document.

- The service has a single Line *Ln_1* with a Line Name of '*A1'*.
- The service is presented in a matrix of five rows of stops (*S_1 – S_5*), and two columns of journeys (#1 – #2), each column showing a vehicle journey stopping at each row.
- There is one **route** (*R_1*), with a single **route section** (*RS_1*) of four **route links** (*RL_1, RL_2, RL_3, and RL_4*). Each route link has two stop references (*RL_1a, RL1b*, etc).
- The service is made up of a single **journey pattern** (*JP_1*). The journey pattern, section and timing links correspond to those of the route; there is a single **journey pattern section** (*JS_1*), and four **timing links** (*JL_1, JL_2, JL_3, JL_4*), with individual **run times** of *10, 20, 8*, and *10* minutes respectively. \There is also a 5 minute wait at sinister street.
  - Each **journey pattern timing link** has two **stop usages** (*JL_1a, JL_1b*, etc), one for each end of the link, i.e. on for departure, one for arrival. These can hold information about the use of the stop
- There are two **vehicle journeys** (*VJ_1, VJ_2*), that both use the same journey pattern, and that are for the same line, '*A1*' (*Ln_1*).
  - For *VJ_1,* each of the four vehicle journey timing links (*VL_1, VL_2, VL_3, VL_4*) corresponds to a link of the journey pattern, and has its own pair of **stop usages** (*VL_1a, VL_1b*, etc).
  - Times at each stop are computed from the vehicle journey **start time** (e.g. '*8.02'*) and the individual link **run times** (e.g. +10mn), plus any **wait time** on the stop usage. (For *S_1 – S_4*, only departure times are actually shown in *Table 3-5***;** for *S_5* it is the arrival time).
  - The second vehicle journey *VJ_2* reuses the links of the first journey pattern *VJ_*, with a different start time ('*10:02'*).

					Journeys	
SV_1	**Service**				**#1**	**#2**
			Line		*Ln_1*	*Ln_1*
	Route	**Journey Pattern**			*JP_1*	*JP_1*
	R_1	*JP_1*	**Vehicle Journey**		*VJ_1*	*VJ_2*

	Route Section		Section		*VJ_1*		*VJ_2*	Name\Line	**A1**	**A1**
Stop	*RS_1*		*JPS_1*		*8:02*		*10:02*			
	Link	Ref	Link	Usage	Link	Usage	Link			
S_1	*RL_1*	*RL_1a*	*JL_1*	*JL_1a*	*VL_1.*	*VL_1a*	←	**Grub Street**	8:02	10:02
S_2		*RL_1b*	*+10mn*	*JL_1b*		*VL_1b*		**Tin Pan Alley**		
	RL_2	*RL_2a*	*JL_2*	*JL_2a*	*VL_2*	*VL_2a*	←		8:12	10:12
S_3		*RL_2b*	*+20mn*	*JL_2b Wait +5mn*		*VL_2b*		**Sinister Street**		
	RL_3	*RL_3a*	*JL_3*	*JL_3a*	*VL_3*	*VL_3a*	←		8:37	10:37
S_4		*RL_3b*	*+08mn*	*JL_3b*		*VL_3b*		**Swans Way**		
	RL_4	*RL_4a*	*JL_4*	*JL_4a*	*VL_4*	*VL_4a*	←		8:45	10:45
S_5		RL_4b	*+10mn*	JL_4b		VL_4b		**Howard's End**	8:55	10:55

Table 3-5 – Structure Example of Schedule: Shared Journey Pattern

3.6.2 Structure Example of a Schedule with an Express Journey

As a slight variation on the structure example given above, we consider a second example (*Table 3-6*), in which the second vehicle journey (*VJ_3*) omits a particular stop *(S_2*) in the same journey pattern (*JP_1*).

- The second journey declares its own distinct set of vehicle journey timing links (*VL_3_1, VL_3_2, VL_3_3, and VL_3_4*) for the journey, so that it can modify the activity. These are based on the same journey pattern.
- For the stop that is omitted (*S_2*), an override value of 'pass' is specified for the activity on the vehicle journey stop usage of the link ends which connect to the stop (*VL_3_1b, VL_3_2a*).

									Journeys	
SV_1	**Service**								**#1**	**#2**
					Line				*Ln_1*	*Ln_1*
	Route		**Journey Pattern**						*JP_1*	*JP_1*
	R_1		*JP_1*		**Vehicle Journey**				*VJ_1*	*VJ_3*
	Route Section		**Section**		*VJ_1*		*VJ_3*	Name\Line	**A1**	**A1**
Stop	*RS_1*		*JPS_1*		*8:02*		*10:02*			
	Link	Ref	Link	Usage	Link	Usage	Usage			
S_1	*RL_1*	*RL_1a*	*JL_1*	*JL_1a*	*VL_1_1*	*VL_1_1a*	*VL_3_1a*	**Grub Street**	8:02	10:02
S_2		*RL_1b*	*+10mn*	*JL_1b*		*VL_1_1b*	*VL_3_1b*	**Tin Pan Alley**		pass
	RL_2	*RL_2a*	*JL_2*	*JL_2a*	*VL_1_2*	*VL_1_2a*	*VL_3_2a*		8:12	pass
S_3		*RL_2b*	*+20mn*	*JL_2b*		*VL_1_2b*	*VL_3_2b*	**Sinister Street**		
	RL_3	*RL_3a*	*JL_3*	*JL_3a*	*VL_1_3*	*VL_1_3a*	*VL_3_3a*		8:32	10:32
S_4		*RL_3b*	*+08mn*	*JL_3b*		*VL_1_3b*	*VL_3_3b*	**Swans Way**		
	RL_4	*RL_4a*	*JL_4*	*JL_4a*	*VL_1_4*	*VL_1_4a*	*VL_3_4a*		8:40	10:40
S_5		*RL_4b*	*+10mn*	*JL_4b*		*VL_1_4b*	*VL_3_4'b*	**Howard's End**	8:50	10:50

Table 3-6 – Structure Example of Schedule: Express VehicleJourney

3.6.3 Plotting a route on a Map

If *Track* data is present it can be used to plot an exact route track on a map. In this case the *Mapping* data should be regarded as independent of the stop locations. That is to plot a route the last point of each mapping is connected to the first point of the succeeding Mapping*.* Thus the track data may follow the centreline of the road.

3.7 Inheriting Timing Link Values

Table 3-7 shows the various values that may be specified for the ***VehicleJourney*** and ***VehicleJourneyTimingLink*** elements, and whether they are:

(i) **Required** [‘*R*’].
(ii) **Optional but otherwise inherited** from the previous level of discourse [‘*O*’].
(iii) **Always** inherited [‘*I*”].
(iv)

The most significant properties are the actual run and wait times of each timing link, but several other operational values, such as fare stages, may also be specified.
For elements that are optional at all levels, a default value is identified to use if no explicit value is provided. For some ‘successive’ properties, such as fare stage number, the value in effect from any previous link is assumed unless specified otherwise. This is indicated by a [‘*S*’].

		Level of Discourse					
Level	**Property**	***Service***	***Route***	***Journey Pattern***	***Vehicle Journey***	**Default Value**	**TXC v**
Pattern	***ServiceRef***		--	(R)	I		2.1
	Direction	O	**--**	O	O	Outbound	2.1
	OperatorRef	**R**	--	O	O	Service / RegisteredOperator)	2.1
	DestinationDisplay	(**R**)	--	O	O	Service /Destination	2.1
	TicketMachineServiceCode	O	--	O	O	none	2.1
	TicketMachine / JourneyCode		--	O	O	none	2.1
	TicketMachine / Direction		--	O	O	Direction	2.1
	Block / Board		--	O	O	none	2.1
	Block / BoardNumber		--	O	O	none	2.1
	Block / Note		--	O	O	none	2.1
	GarageRef		--	O	O	none	2.1
	VehicleType	O	--	O	O	none	2.1
	AssistanceService	O	--	O	O	none	2.5
	ServiceFacilitySet	O	--	O	O	none	2.5
	LayoverPoint		--	O	O	none	2.1
	TimeDemand		--	O	O	none	2.4
	CommercialBasis	O	--	O	O	unknown	2.1
	Frequency		--	O	O	false	2.1
	OperatingProfile	O	--	O	O	Monday to Friday, Every Day of Year	2.1
	LineRef	O	--	--	**R**	--	2.1
	DepartureTime		--	--	**R**	--	2.1
Section	order		--	O	I	None	2.1
↓							2.1
TimingLink	***LinkRef***		--	O	**R**	--	2.1
	Direction	O	**R**	O	I	JourneyPattern / Direction	2.1
	RunTime		**--**	**R**	O	--	2.1
							2.1
	Distance		O	O	I	zero	2.1
	DestinationDisplay		--	O	O	none (same as Pattern / DestinationDisplay)	2.1
	Vias /ViaName		--	O	O	none	2.1
	HailAndRide		O	O	O	false	2.1
	DutyCrewCode		--	O (S)	O (S)	none	2.1
	CommercialBasis	O	--	O	O	Same as pattern	2.1
	StoppingArrangements		--	O	O	none	2.1
↓							2.1
TimingLink StopUsage	***StopPointRef***		**(R)**	**R**	I	--	2.1
	TimingStatus		**--**	O	I	TIP	2.1
From & To	***Activity***		--	O	O	PickUpAndSetDown	2.1
	WaitTime		(++)	O	O	zero	2.1
	VariableStopAllocation			O	O	none	2.1
	FareStageNumber		--	O (S)	I	none	2.1
	FareStage		--	O	I	false	2.1

Table 3-7 – Journey Properties and Defaults

++ A default wait time may be specified on stops. This merely sets a default that may be used to set the initial value used by services. Each journey pattern sets the wait value on each timing link.

Figure 3-25 shows how inheritance relationships are used in the *TransXChange* supply model so as to express the shared attributes and common data structure of equivalent elements, that is, the elements and subelements of ***JourneyPattern*** and ***VehicleJourney***. For each element type, an abstract class is used to represent the common properties, and distinct subtypes describe any specific differences. For example, ***AbstractTimingLink*** has subtypes ***JourneyPatternTimingLink*** and ***VehicleJourneyTimingLink***.

See Table 3-7 for attributes which may be inherited.

- A **VehicleJourney** may override any common property it shares with a **JourneyPattern.**
- A ***VehicleJourneyTimingLink*** may override any common property it shares with a ***JourneyPatternTimingLink*.**
- A ***VehicleJourneyStopUsage*** may override any common property it shares with a ***JourneyPatternStopUsage*, including the *StopAccessibility*.**

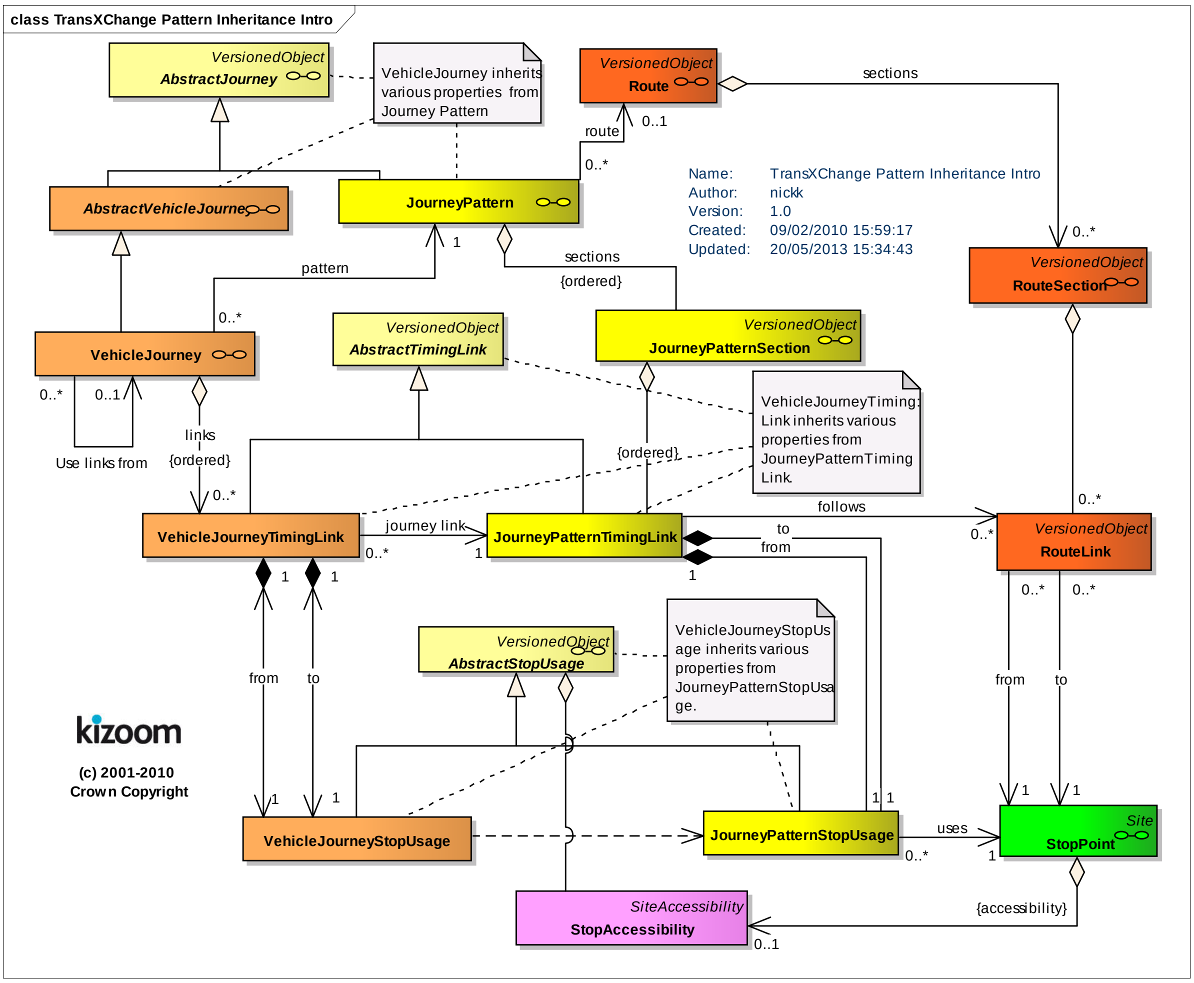


Figure 3-25 – UML Diagram of Service Pattern elements

3.7.1 Schedule and Journey Terms and Definitions

The *TransXChange* uses the following definitions of common scheduling terms. See also the definitions of individual schema elements. Some of the terms are used in actually element names; others merely define concepts.

3.7.1.1 Time Related Terms

- *Relative time:* Time as a duration, usually in minutes, for example, '*5 minutes'*.
- *Absolute time:* Time as a specific clock hour, for example, '*10:00*', '18:30'.
- *Overall Wait time:* Relative time to wait at a specific stop, assuming bus arrives on time. Used to compute passing times. In real-time operations, if bus is late at a stop, wait time may be reduced to the minimum time need to disembark and board passengers, i.e. wait is a buffer time used to adhere to schedule. The actual time waited is the *Dwell time* - which is an operational time and not relevant to *TransXChange.* Note that wait time is a property of a journey pattern or vehicle journey, not of the stop itself, since it may be different on different journeys using the same stop. In *TransXChange,* the overall wait time is computed from two separate component timing link wait times that can be stated on each end of the incoming and outgoing ***JourneyPatternTimingLink*** or ***VehicleJourneyTimingLink*** instances:
    - See ***JourneyPatternStopUsage / WaitTime.***
    - See ***VehicleJourneyStopUsage / WaitTime***.
- *Run time.* Relative time taken to traverse a timing link.
    - See ***JourneyPatternTimingLink / RunTime.***
    - See ***VehicleJourneyTimingLink / RunTime***.
- *Departure Time:* The absolute time at which a vehicle journey leaves from its first stop.
    - See ***VehicleJourney / DepartureTime***.
- *Passing time:* Absolute time that a bus reaches a stop. Comprises the departure time from the previous stop, plus the run time for the timing link connecting the previous stop and the next stop. Derived.
- *Frequency Based Service:* A service that runs to a regular frequency, for example '*every 5 minutes'*, rather than to a specific timetable. May or may not be a strict Frequent Service.
    - See ***VehicleJourney / Frequency***.
- *Frequent Service*, a service that runs to a frequency of every 10 minutes or less in accordance with the Statutory Requirement, and that has been formally registered as constituting a *Frequent Service*. Normally, but not necessarily, a *Frequency Based Service.*
    - See ***VehicleJourney / Frequency / FrequentService***.
- *Day Type:* A type of day or day such as Monday, Weekday, or Weekend as opposed to a calendar date.

3.7.1.2 Routing Related Terms

- *Block:* A description of a group of journeys to be operated by a particular vehicle, in a specific working period, normally covering a full working day. May be identified by a *block number*.
    - See ***JourneyPattern / Block / Description***.
    - See ***JourneyPattern / Block / BlockNumber***.
- *Origin:* The place from which the service starts. Does not vary; note however that some journeys of the service may have a '*short working'* so start from a different actual origin when executed.
    - See ***Service*** / ***Origin***.
- *Destination:* The place to which the service goes. Does not vary; Note however that some journeys of the service may have a 'short working' and so go to a different actual destination when executed.
    - See ***Service*** / ***Destination***.
- *Destination Display:* Name of a destination to which the bus ultimately goes. Fixed for whole journey.
    - See ***JourneyPattern / DestinationDisplay.***
    - See ***VehicleJourney / DestinationDisplay***.
- *Dynamic Destination Display:* Name of a destination where the bus is currently considered heading, shown on the front of the bus. Also known as the *Heading.* On a circular or other route with a complex topology, the destination display may change from stop to stop. On a

linear route, normally the same as the destination display, but on a short working may be an earlier point in the pattern.
  - See ***JourneyPatternTimingLink / DestinationDisplay.***
  - See ***VehicleJourneyTimingLink / DestinationDisplay***.
- *Stop List.* The actual list of stops at which the bus will stop, in order of visiting. Sometimes also termed the '*calling pattern'*.
- *Via List:* The list of place names that is published for the service. This may be a subset of the stop list and may include names that are not stops
- *Direction*: relative course of a bus following a vehicle journey – may be outbound, inbound, clockwise or anti-clockwise.
  - ***JourneyPattern / Direction, JourneyPatternTimingLink / Direction***.
  - See ***VehicleJourneyTimingLink / Direction***.
- *Bearing*, Absolute, i.e. compass direction of a bus along a street, e.g. 'North'.
  - See ***StopPoint / Bearing***.
- *Layover Point:* Point at which a bus may stop and wait until it is time to start the next service stage.
  - See ***JourneyPattern / LayoverPoint***.
- *Short Working:* A vehicle journey that follows a journey pattern but omits one or more stops at one or other end of the journey.
  - See ***VehicleJourney / DeadRun / EndStopUsage***.
- *Express Journey:* A vehicle journey that follows a journey pattern but passes certain stops without stopping (also referred to as a *Limited Stop Journey*).
  - See ***JourneyPatternTimingLink / Activity.***
- *Stop Footprint*: The geometry of the stop coverage. Most stops are points. Some stop types however have a footprint that covers more than a single point, for example hail and ride sections, or flexible zones.

3.7.2 Computation of Passing Times

The passing time at each stop (see *Figure 3-26*) is calculated from the cumulative sum of the individual timing link values for all preceding stops in the journey link sequence as follows:

[1] Arrival time at stop$_n$ = Departure time from previous stop $_{n-1}$ + (Run time for inbound link from stop $_{n-1}$)

[2]. Departure time at stop$_n$ = Arrival time at stop$_n$ + Wait time for destination end of inbound link from stop$_{n-1)}$. + Wait time for origin of outbound link to stop $_{n+1}$

Where:

1. Default **vehicle journey wait** times for each link are derived from the journey pattern timing link onto which the vehicle journey timing link projects (i.e. through the ***VehicleJourneyTimingLink / JourneyPatternTimingLinkRef),*** as follows:
   - If no value for wait time is specified on the *departure* end of the timing link, i.e. for the ***VehicleJourneyTimingLink*** / ***From*** / ***VehicleJourneyStopUsage***, the default ***WaitTime*** from the corresponding ***JourneyPatternTimingLink*** / ***From*** / ***JourneyPatternStopUsage*** is used.
   - If no value for wait time is specified on the *arrival* end of the timing link, i.e. the ***VehicleJourneyTimingLink*** / ***To*** / ***VehicleJourneyStopUsage***, the default ***WaitTime*** from the corresponding ***JourneyPatternTimingLink*** / ***To*** / ***JourneyPatternStopUsage*** is used.

2. If unspecified, journey pattern **wait** times are defaulted as follows:
   - If no value for wait time is specified on the *departure* end of the timing link, i.e. the ***JourneyPatternTimingLink*** / ***From*** / ***JourneyPatternStopUsage***, a value of **zero** is used.

- If no value for wait time is specified on the *arrival* end of the timing link, i.e. the ***JourneyPatternTimingLink*** / ***To*** / ***JourneyPatternStopUsage***, a value of **zero** is assumed.

3. Default vehicle journey **run** times for each link are derived from the journey pattern timing link onto which the vehicle journey timing link projects. A run time is mandatory on each ***JourneyPatternTimingLink.***

The structured example shown earlier gives a simple example of how passing times are derived from run times and wait times.

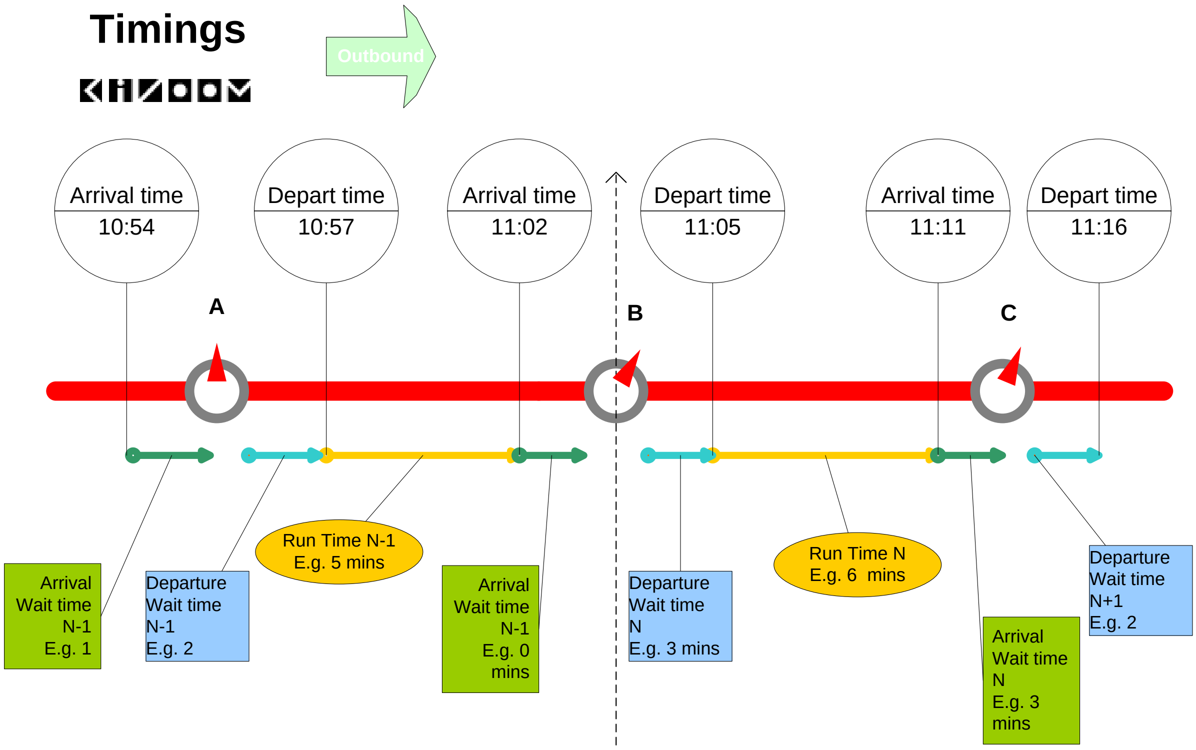


Figure 3-26 – Computation of Passing Times

3.7.2.1 Example of Inheritance of Passing Times

Table 3-8 shows a more complex example, where wait and run times are specified at different levels of discourse, that is, default values from the journey pattern are used except where overridden by the vehicle journey. For each step, the wait and run times are added to values from the previous step to arrive at an overall passing time. There are three stops S1, S2, S3 and two links (L1. L2) between them.

- An initial time of '*10:00'* is specified.
- Run time *R1 (*5 minutes*)* on the vehicle journey pattern timing link (L1) is defaulted from the journey pattern timing link.
- Run time *R2 (*10 minutes*)* on the vehicle journey pattern timing link (L2) overrides the default (14 minutes) on the journey pattern.
- Departure wait time *W1b* at S1 (2 minutes) on the vehicle journey timing link end L1a overrides the default (0 minutes) on the journey pattern.
- Arrival Wait time *W2a* (5 minutes) at S2 on the vehicle journey timing link end L1b is defaulted from the journey pattern.
- Departure wait time *W2b* at S2 (7 minutes) on the vehicle journey timing link end L2a overrides the default (6 minutes) on the journey pattern.
- Arrival Wait time *W3a (*10 minutes*)* at S3 on the vehicle journey timing link end L2b overrides the default (5 minutes) on the journey pattern.
- Departure wait time *W3b* at S3 (5 minutes) - which would come from a successor link L3) can be used to compute the departure time from S3.

•

Stop			Link							Usage			
			Wait Time			Run Time				Computation		Passing Time	
	T		JP	VJ	Act-ual	id	JP mns	VJ mns	Actual mns		T		Actual
											t1		10:00
s1	-	**-**	--	--	*0*					-		**a**	
	w1b	**From**	*(+0)*	--	*+2*	**L1a**				*t1 + w1b*	*t2*	**d**	10:02
	r1					**L1**	*+5*	--	*+5*	-			
s2	*w2a*	**To**	*+5*	--	*+5*	**L1b**				*t2 + r1*	*t3*	**a**	10:07
	w2b	**From**	*(+6)*	*+7*	*+7*	**L2a**				*t3 + w2a + w2b*	*t4*	**d**	10:19
	r2					**L2**	*(+14)*	*+10*	*+10*				
s3	*w3a*	**To**	*(+5)*	*+10*	*+10*	**L2b**				*t4 + r2*	*t5*	**a**	10:29
	w3b	**From**	--	*+5*	*+5*	**L3a**				*t5 + w3a + w3b*	*t6*	**d**	10:34
						L3	.	.	.				
		To.				**L3b**							

Table 3-8 – Example of Computation of Inherited Passing Times

3.7.2.2 Rounding of Passing Times

Run and wait times are specified as values of XML type ***Duration***, which may include seconds, for example *PT10M55S*. The TransXChange publisher computes departure times using the full value including seconds, but in the matrix timetable rounds down the total cumulative time to the nearest whole minute, i.e. the rounded value is not used to reset the cumulative time. Table 3-9 gives an example.

Stop	Run Time	Cumulative Time	Show As
A		7:00:00	7:00
B	PT20M50S	7:20:50	7:20
C	PT20M50S	7:41:40	7:41
D	PT10M55S	7:52:35	7:52

Table 3-9 – Example of Rounding of Passing Times

3.8 Standard Services

3.8.1 Introduction to the Service Model

Figure 3-27 introduces the *TransXChange* ***Service*** model which groups the journeys of a TransXChange document and associates them with common properties. Each ***Service*** can be made up of ***StandardService*** and ***FlexibleService*** components.

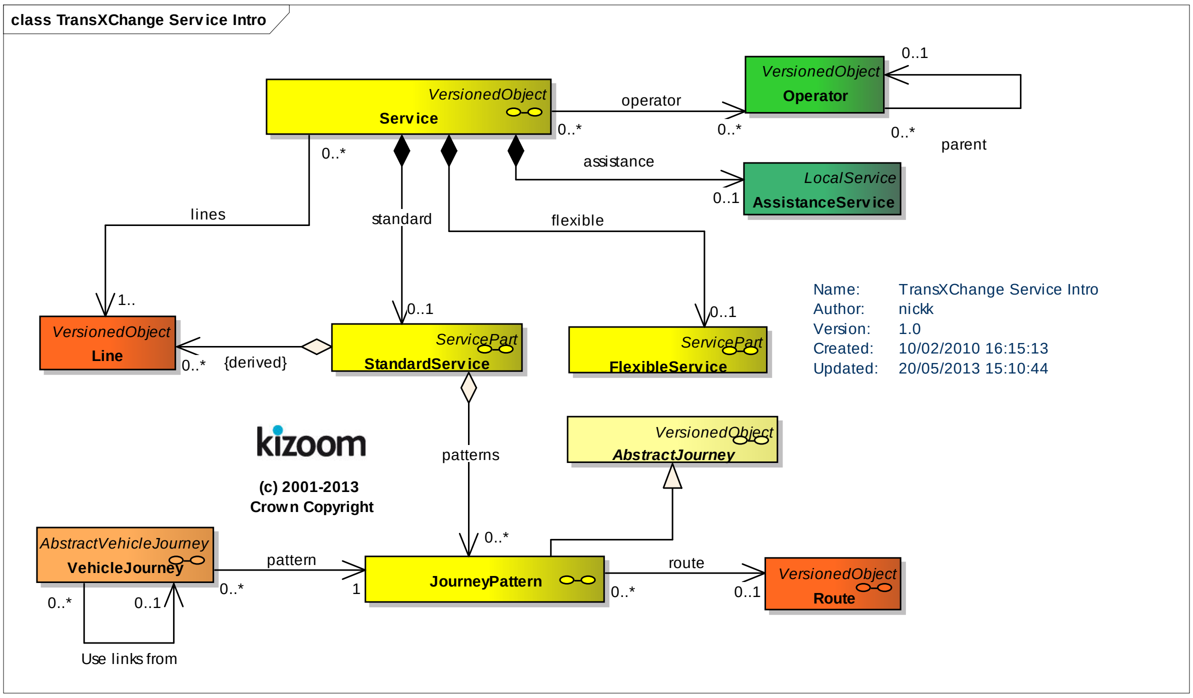


Figure 3-27 – UML Diagram of Service model: Introduction

3.8.2 Introduction to the Standard Service Model

Figure 3-28 gives a slightly more detailed view of the ***Service*** *TransXChange* model introduced above, summarising the overall structure of a ***StandardService***, and showing again that ***Journey-Pattern,*** and ***VehicleJourney*** are made up of collections of timing links (***JourneyPatternTiming-Link,*** and ***VehicleJourneyTimingLink*** respectively), which hold the details about each individual step between stops of the journey.

- Each timing link has information about the arrival and departure of the vehicle at a stop, specified with a stop usage element (***JourneyPatternStopUsage,*** and ***VehicleJourney-StopUsage*** respectively).

- For Bus Stations, stop i.e. bay allocation may be variable, specified by a ***VariableStop-Allocation***.

A ***StandardService*** describes the fixed route component of a ***Service***.

- Each ***Service*** can have one or more ***Line*** instances associated with it; this specifies a label to be associated with journeys, for example, “N93”.

- Each ***StandardService*** must have one or more ***JourneyPattern*** instances.
  - A ***JourneyPattern*** instance may reference a Route and a Track.

- The ***StandardService*** must have one or more ***VehicleJourney*** instances. Each ***Vehicle-Journey*** instances must reference a ***JourneyPattern*** of the same ***StandardService,*** and a ***Line*** instance of the ***Service*** to which it belongs.

- Each ***VehicleJourney*** must specify a ***DepartureTime***: Frequency based services may also describe a ***Frequency***. See 0 below.

Connections with other services are described by *interchanges*. These are described in *Section 3.10*.

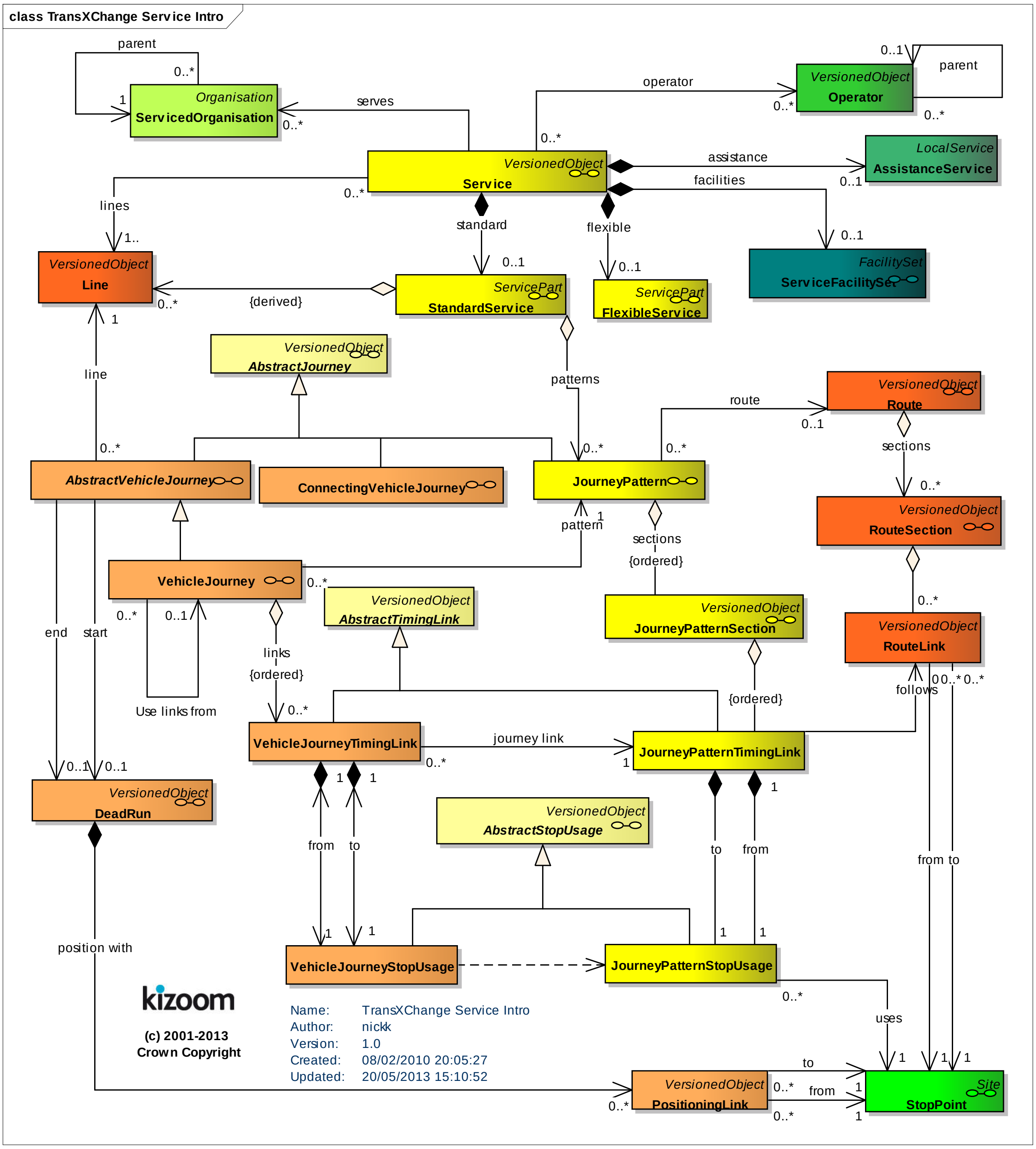


Figure 3-28 – UML Diagram of Standard Service: Overview

3.8.3 Standard Service Properties

Figure 3-29 shows further details of a ***Service*** including a ***ServiceClassification.***

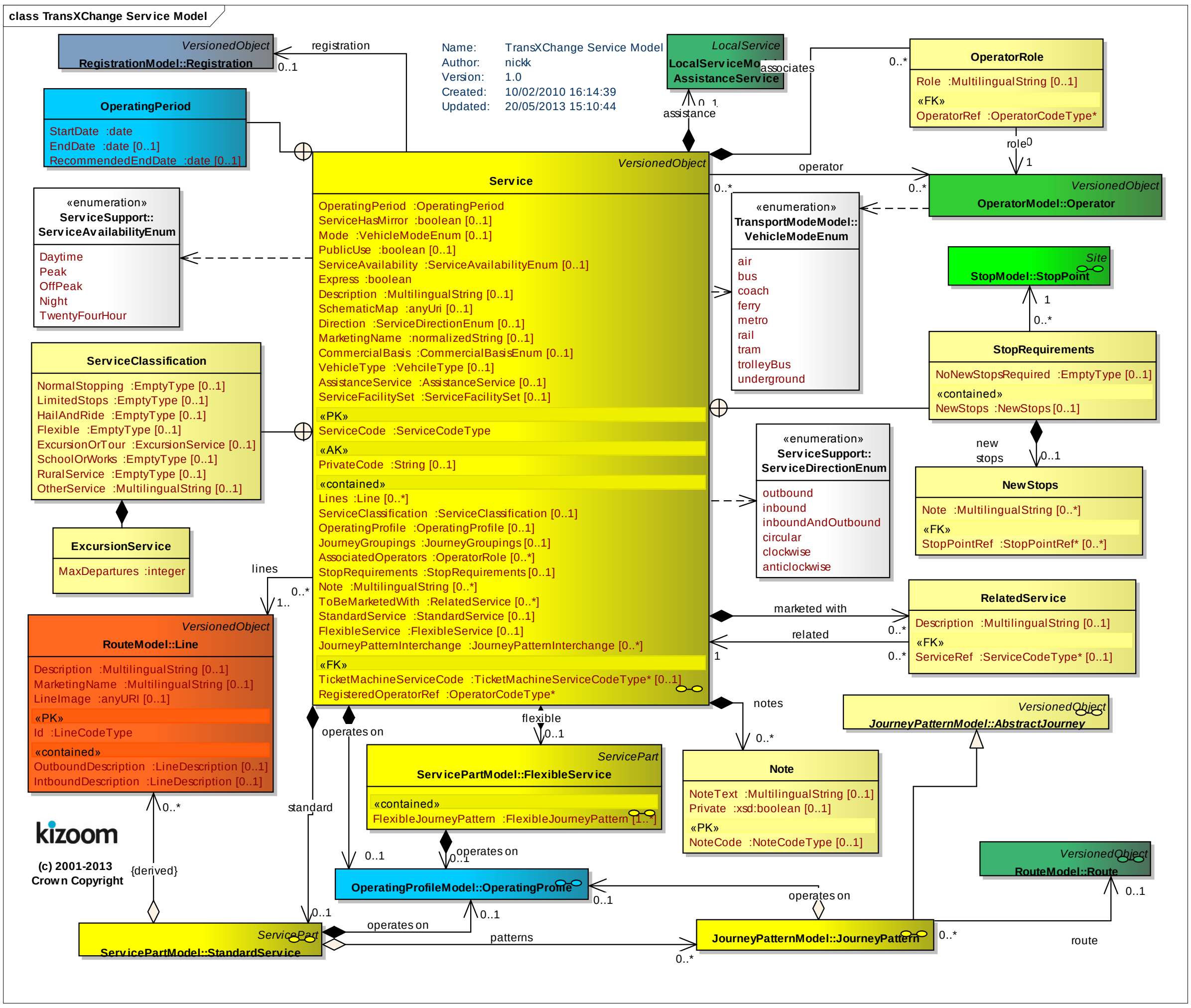


Figure 3-29 – UML Diagram of Standard Service: Details

3.8.4 Standard Service Components

Figure 3-30 shows further details of the ***StandardService*** and ***FlexibleService*** components of a ***Service***.

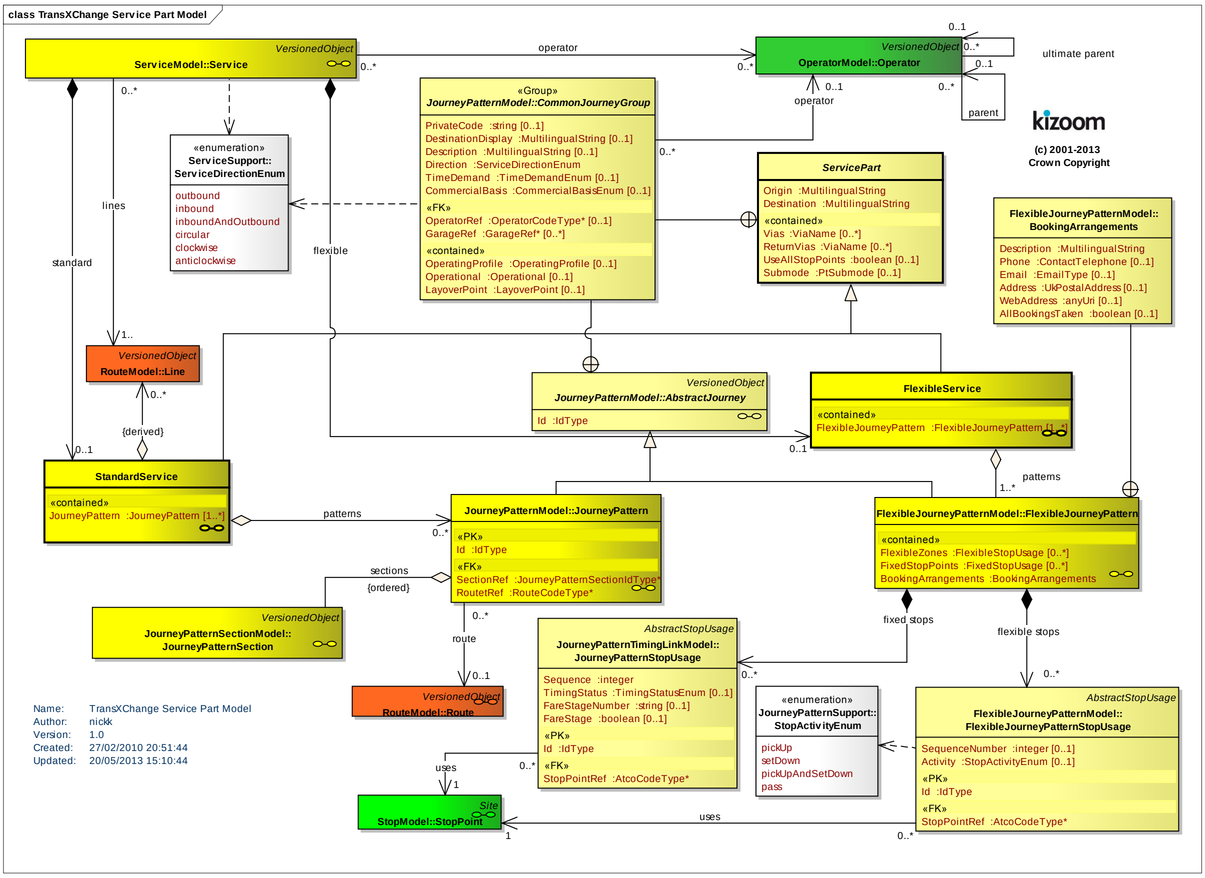


Figure 3-30 – UML Diagram of Standard Service Parts

3.8.5 Service Model Hierarchy

Figure 3-31 shows the inheritance hierarchy for the ***Service*** model elements.

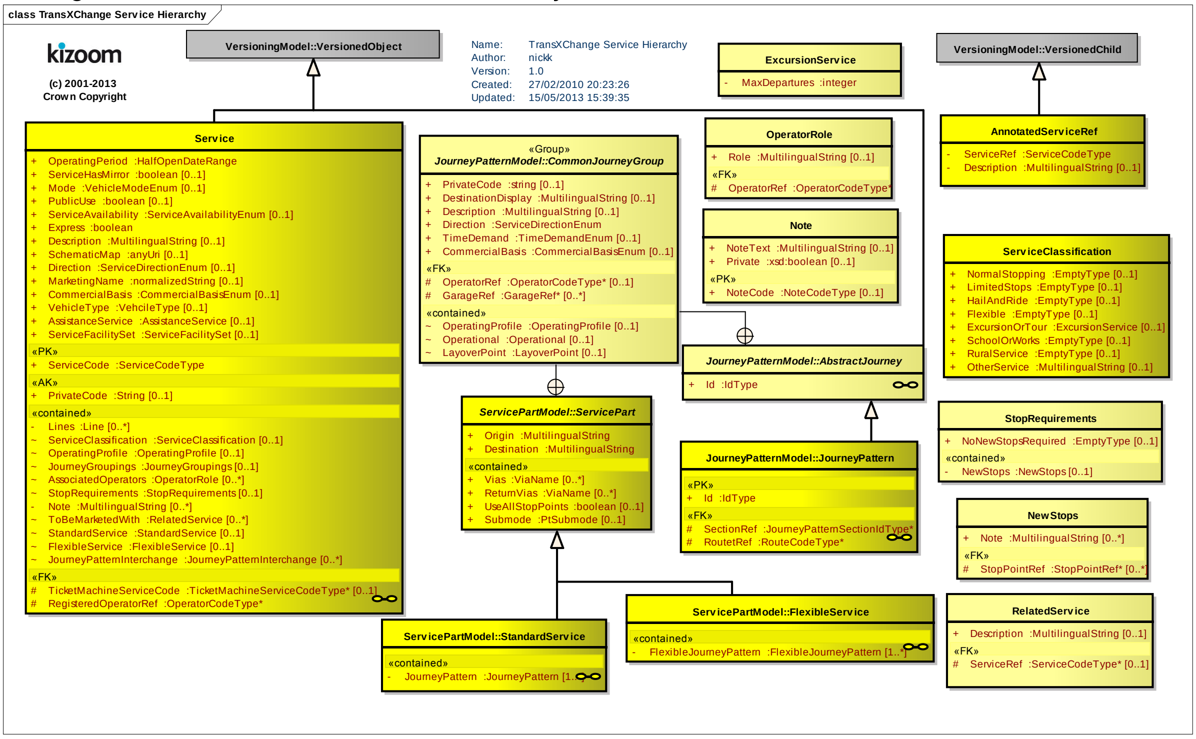


Figure 3-31 – UML Diagram of Service Model Element Hierarchy

3.8.6 Service Model Types

Figure 3-32 shows the additional data types used the ***Service*** model elements.

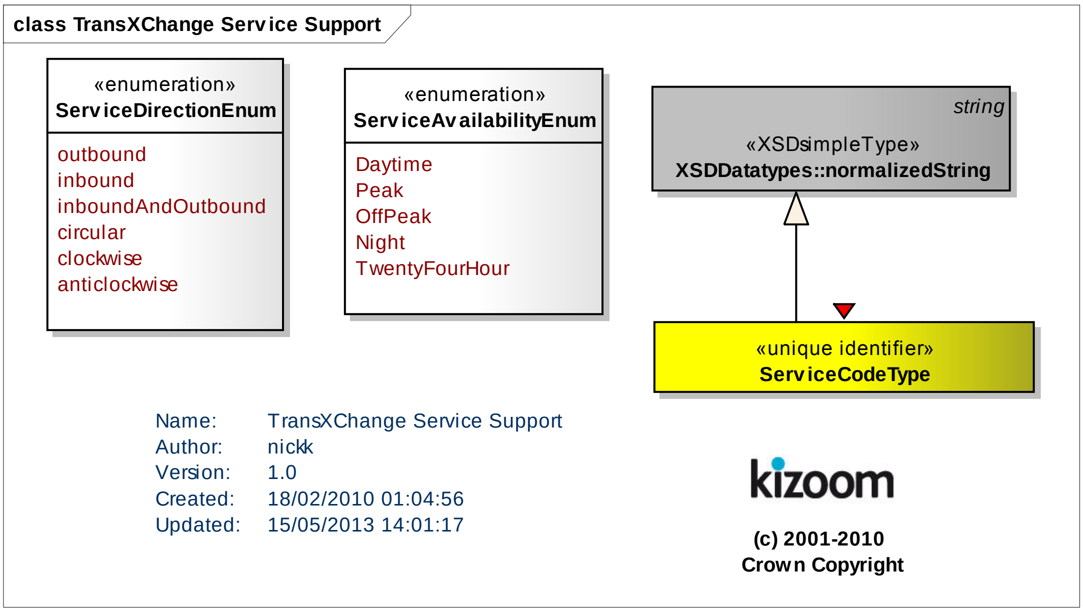


Figure 3-32 – UML Diagram of Service Support types

3.9 Flexibly Routed Services

The *TransXChange* model can also support flexibly routed services (*Figure 3-33*).
A flexible service operates between catchment areas that can be made up of both spatial zones, and lists of fixed stops, allowing combinations of (i) area-to-fixed stop, (ii) area-to-area, (iii) fixed stop-to-fixed stop. Within a zone there is no fixed or marked stop, but the service will call on demand.

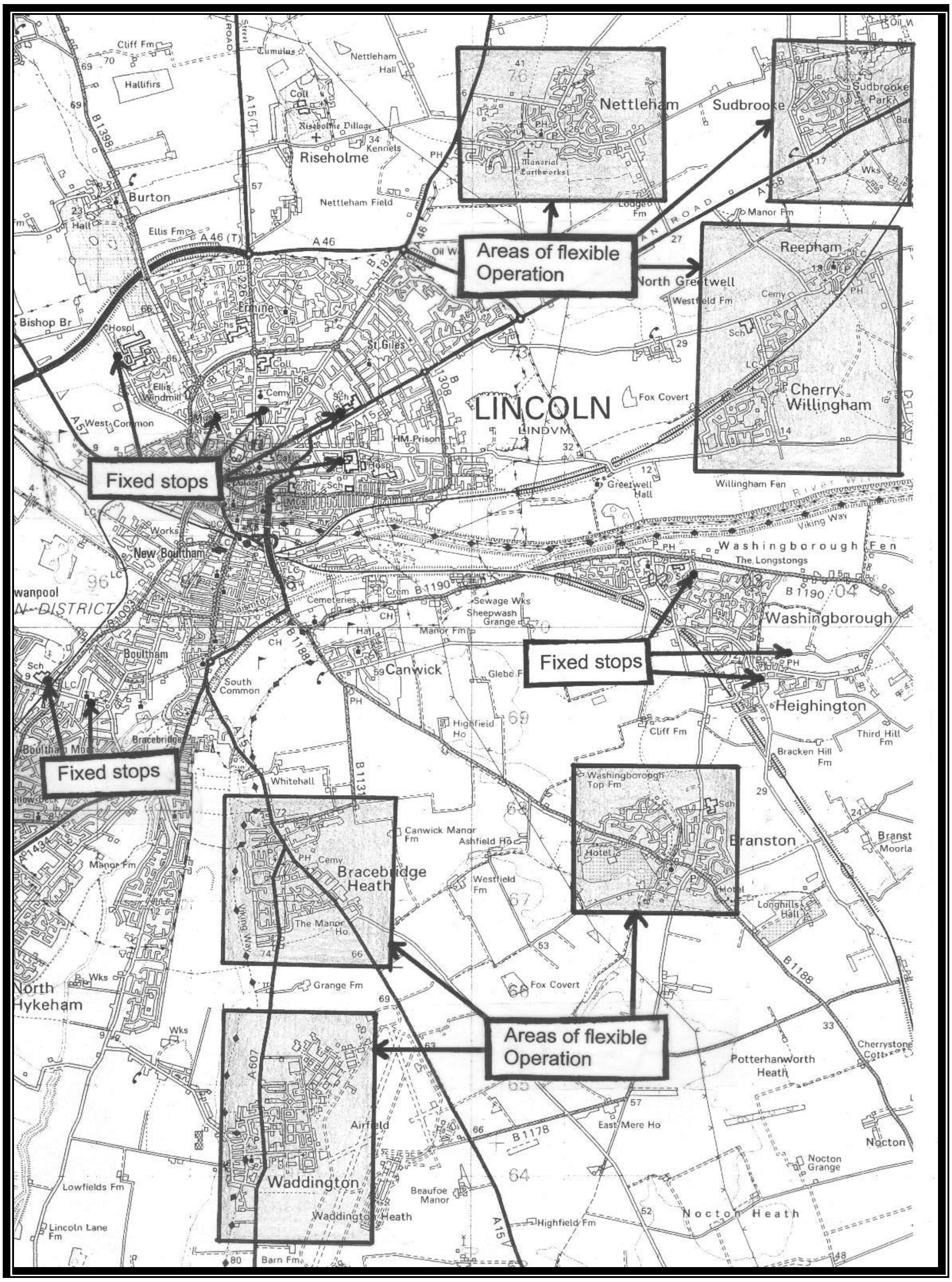


Figure 3-33 – Flexible Network

3.9.1 Introduction to the Flexible Service Model

Figure 3-34 introduces the elements used to represent a ***FlexibleService***.

- A ***FlexibleService*** has a ***FlexibleJourneyPattern,*** which must include some *NaPTAN* stops of type ***FlexibleZone*** (*FLX*) to define areas within which passengers may be picked up or set down.
  - A ***FlexibleZone*** must be a contiguous area. Like other *NaPTAN* stop types, a ***FlexibleZone*** stop can be associated with one or more *NPTG* Localities: the locality with the greatest correspondence to the area of the zone should be used as the primary *NPTG* Locality; other localities that the zone falls within should be specified as alternative *NPTG* localities on the *NaPTAN* stop definition. Where a flexible zone substantially covers two or more NPTG Localities, it is preferable to define two separate zones, one for each locality.
  - A ***FlexibleJourneyPattern*** may also have one or more ***FixedStopPoint*** instances that can be visited in any order by the flexible service. Fixed stops should be *NaPTAN* stops of a type other than ***FlexibleZone*** (*FLX*).
  - The allowed activity (pick up, set down etc) and other behaviour of the service at each stop, fixed or flexible, is defined by a stop usage instance for each stop used.
- A ***FlexibleVehicleJourney*** describes the actual operation of the flexible service, using a ***FlexibleServiceTimes*** element to specify the time bands during which the service operates.
- A ***Service*** may contain both ***FlexibleService*** and ***StandardService*** components. Interchange elements can be used to define the transition between flexible and fixed stages.
- Other properties of the service, such as ***Registration***, ***Operator***, ***Line*** and ***OperatingProfile,*** are specified with the same elements as for a ***StandardService***.

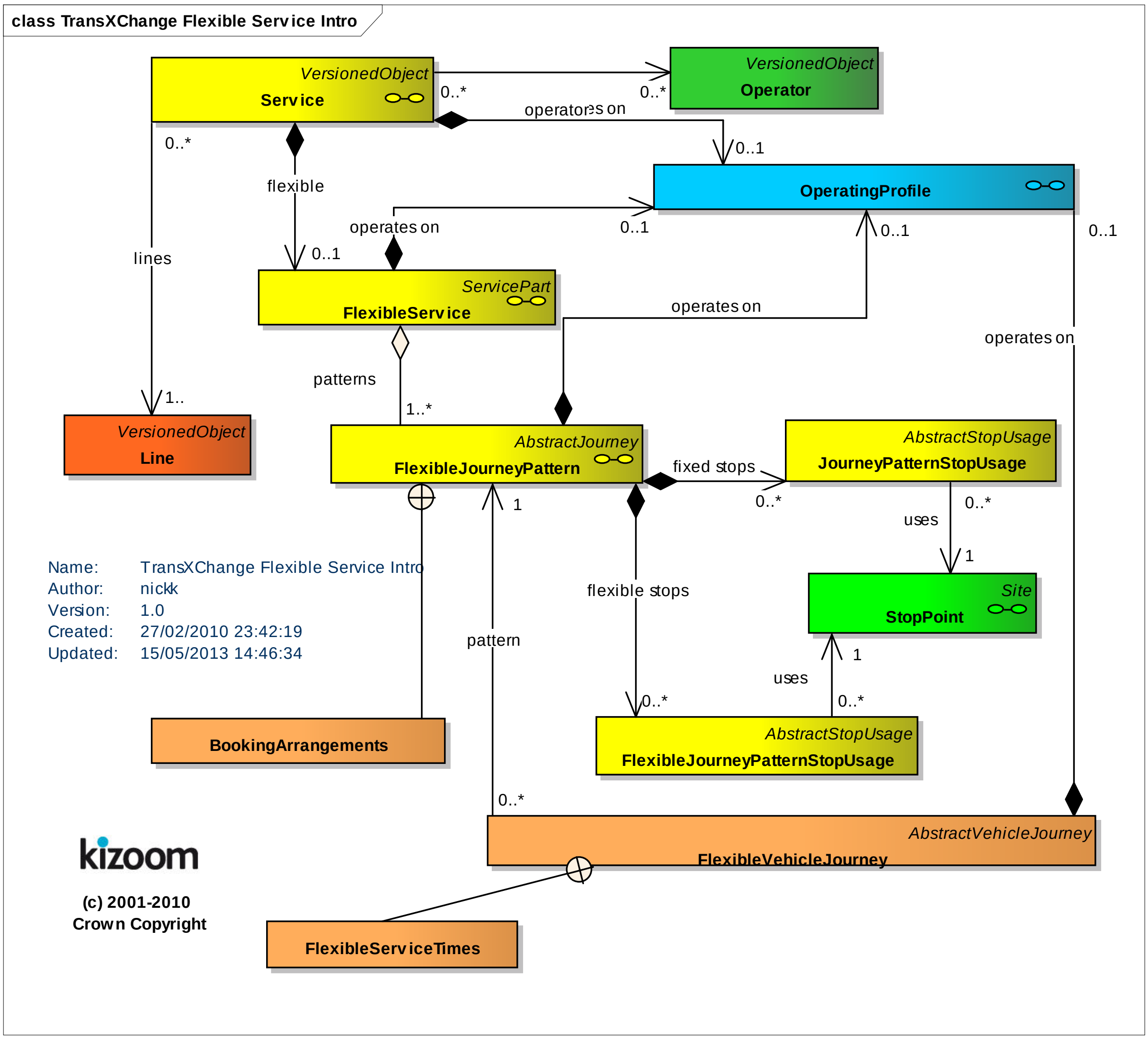


Figure 3-34 – UML Diagram for Flexibly Routed Service: Introduction

3.9.2 Flexible Service Model Details

Figure 3-35 elaborates *Figure 3-34* to show the attributes of Flexible Service elements.

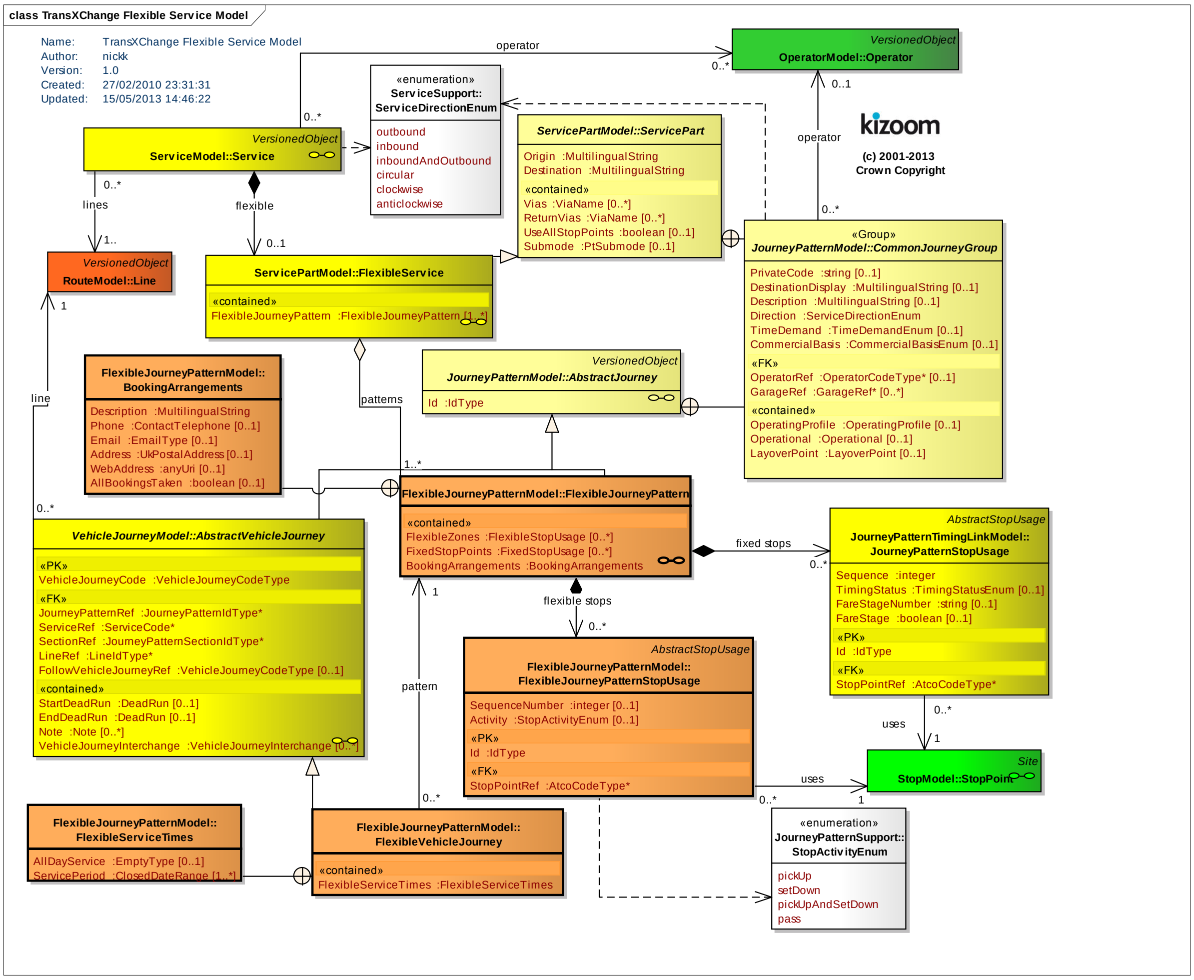


Figure 3-35 – UML Diagram for Flexibly Routed Service: Details

3.9.3 Flexibly Routed Service Hierarchy

Figure 3-36 shows the inheritance hierarchy for the Flexibly Routed Service model elements.

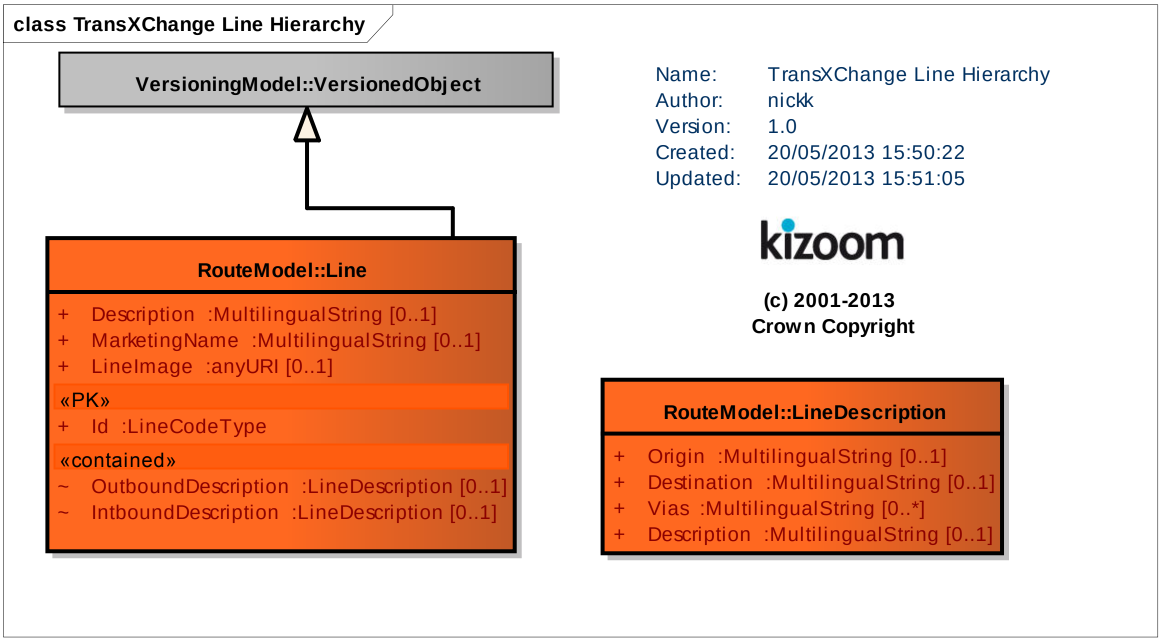


Figure 3-36 – UML Diagram of Flexibly Routed Service Element Hierarchy

3.10 Line Model

3.10.1 Introduction to the Line Model

In TransXChange a Line is an arbitrary grouping of journeys under a common public identifier. Each service may have multiple lines. Figure 3-37 introduces the elements used to represent a ***Line***. These comprise ***Line*** and up to two ***LineDescription*** instances**.** Each ***VehicleJourney*** may reference a ***Line***.

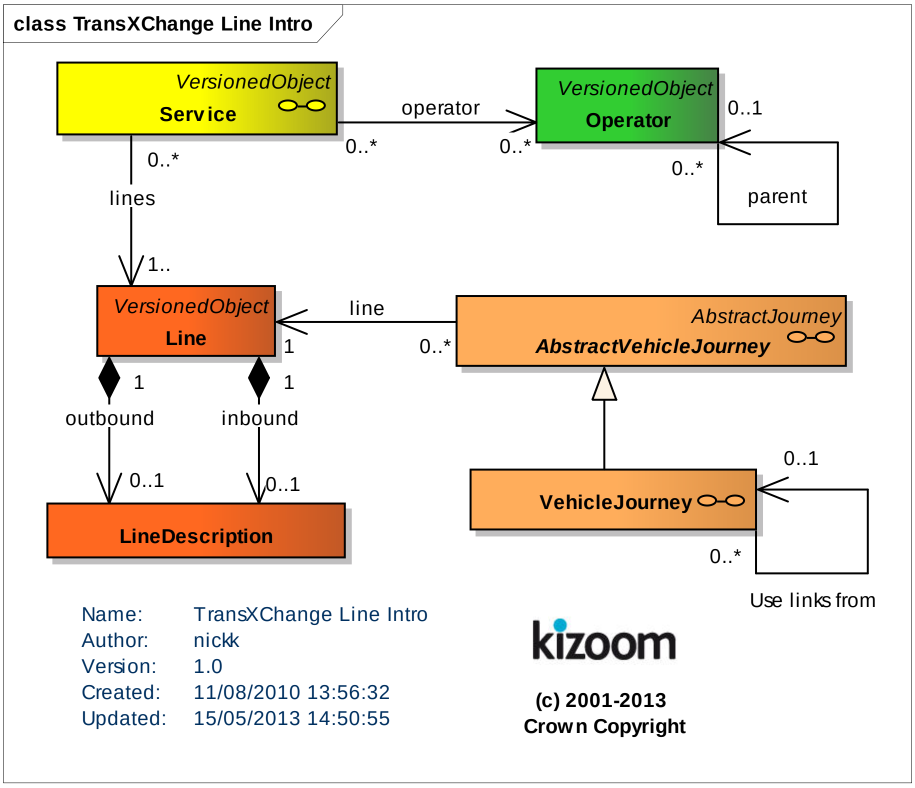


Figure 3-37 – UML Diagram of Line Model: Introduction

3.10.2 Line Model Details

Figure 3-38 elaborates Figure 3-37 to show further attributes.

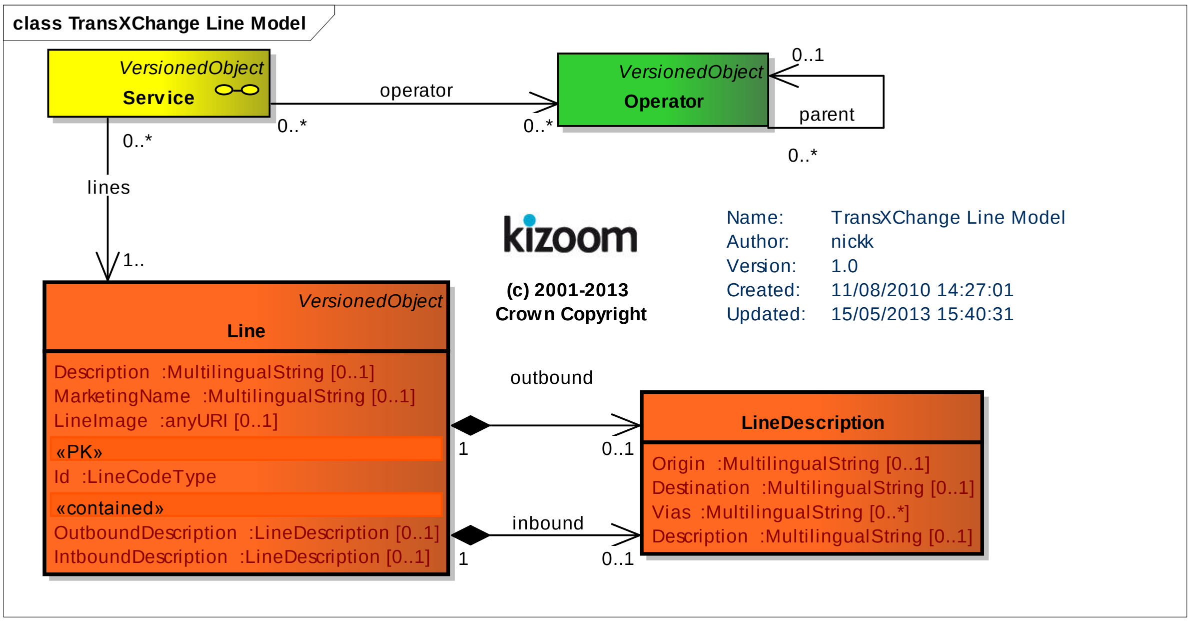


Figure 3-38 – UML Diagram of Line Model

3.10.3 Line Hierarchy

Figure 3-39shows the inheritance hierarchy for the ***Line*** model elements.

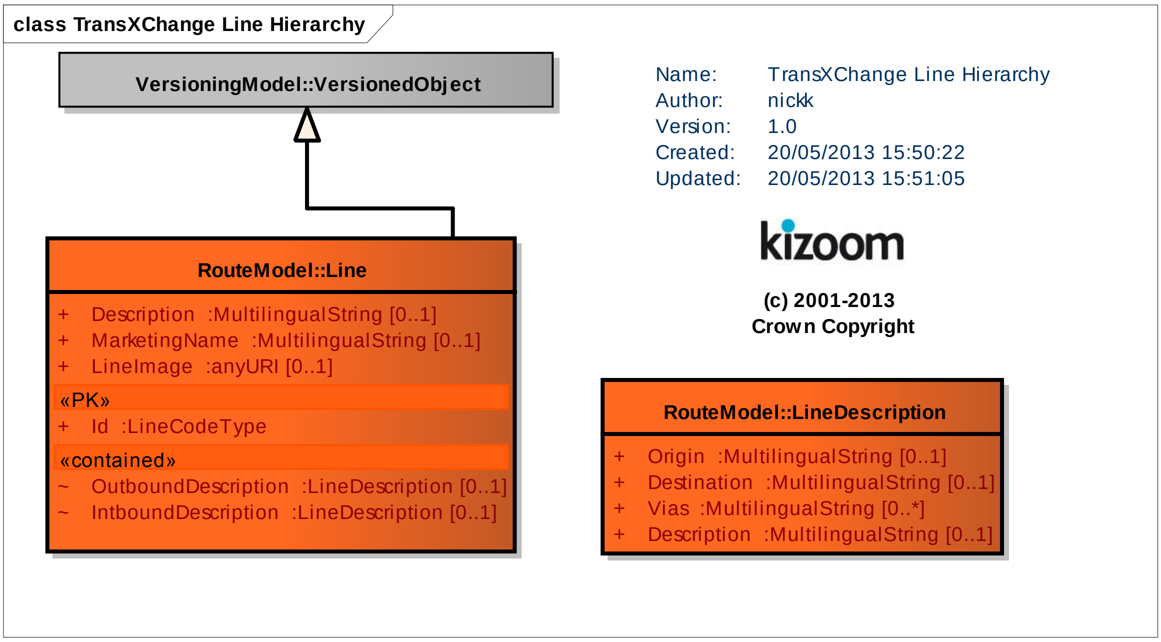


Figure 3-39 – UML Diagram of Line Model Element Hierarchy

3.11 Interchanges

To specify the connection between vehicle journeys, an Interchange model is used, as shown in the UML structure diagram in *Figure 3-40*. The Interchange model operates on two levels of discourse:

- A ***JourneyPatternInterchange*** specifies a possible connection between any two ***JourneyPattern*** instances, at a particular stop or pair of stops, with default values for the connection activity.
  - A service may hold multiple connections.
  - The arrival stop of the inbound 'feeder to' journey, and the departure stop of the outbound 'distributor from' journey may be different *NaPTAN* stop points, i.e. require a transfer.
  - The mode of transfer (e.g. walk or otherwise) is indicated by a ***TransferMode*** property.

- A ***VehicleJourneyInterchange*** specifies the connection between two specific ***VehicleJourney*** (or ***ConnectingVehicleJourney***) instances, at a ***VehicleJourneyInterchange.*** A vehicle journey connection can project onto an equivalent ***JourneyPatternInterchange***, which constrains it to use the corresponding inbound feeder and outbound distributor journey pattern as in the reference, and the same stops specified by the ***JourneyPatternInterchange***.
- A vehicle journey may have connections with more than one other vehicle journey.
- Specification of the connecting ***VehicleJourney*** may be done in either of two ways.
  - Using a normal ***VehicleJourney***, with ***VehicleJourneyTimingLinks*** and full details
  - Using a ***ConnectingVehicleJourney*** – This allows a more lightweight statement of the connecting ***VehicleJourney*** without providing all of its details.

Note that inbound 'feeder to' and outbound 'distributor from' are relative roles; and a given service may serve as both feeder and distributor (i.e. passengers may exchange both ways between vehicles); in which case separate interchange instances can be declared for each direction.

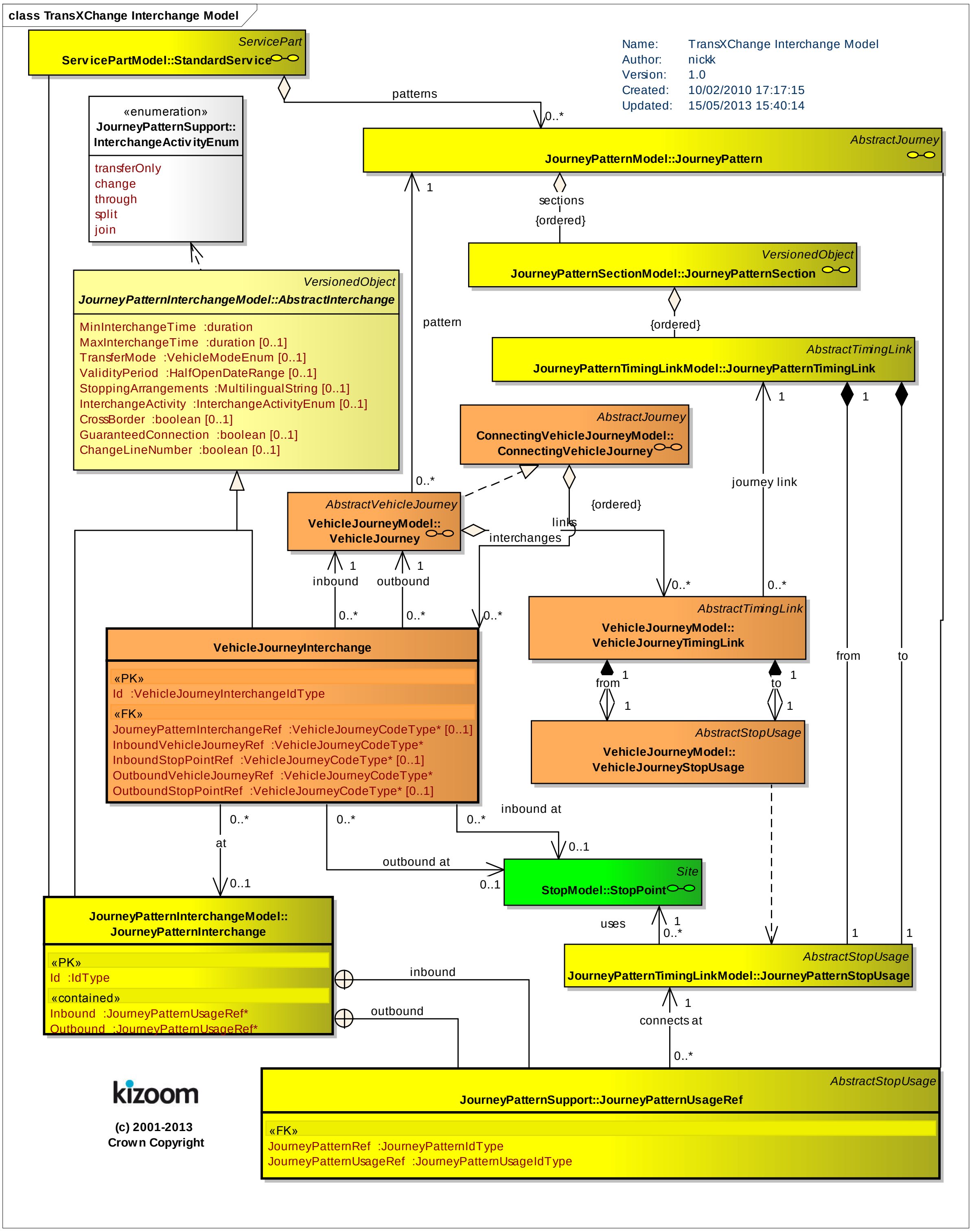


Figure 3-40 – UML Diagram of Interchanges

3.11.1 Inheriting Interchange Values

Table 3-7 shows the various values that may be specified for the ***JourneyPatternInterchange*** and ***VehicleJourneyInterchange*** elements, and whether they are:

(i) **Required** (‘R’).
(ii) **Optional** but otherwise inherited from the previous level of discourse (‘O’).
(iii) Always **Inherited**. (‘I’).

For elements that are optional at all levels, a default value is identified to use if no explicit value is provided.

Level	Property	*Journey Pattern*	*Vehicle Journey*	Default Value
Interchange	***InboundJourneyPatternRef***	**R**	I	--
	OutboundJourneyPatternRef	**R**	I	--
	InboundStopUsageRef	**R**	I	--
	OutboundStopUsageRef	**R**	I	--
	InterchangeActivity	**O**	O	*change*
	MinInterchangeTime	**R**	O	--
	MaxInterchangeTime	**O**	O	*Zero*
	InterchangeMode	O	I	*walk*
	ValidityPeriod	O	O	*service end date*
	StoppingArrangements	O	O	*none*
	JourneyPatternInterchangeRef	**-**	O	
	InboundVehicleJourneyRef	-	**R**	--
	InboundStopPointRef		O	inherit
	OutboundVehicleJourneyRef	-	**R**	--
	OutboundStopPointRef		O	inherit

Table 3-10 – Interchange Properties and Defaults

3.11.2 Interchange Schematic

Figure 17 shows a schematic diagram of an interchange between two journeys. The inbound feeder journey arriving at stop '*A*' from stop '*X*' connects to a second distributor journey from stop '*A*' onto stop '*B*'. The journey pattern interchange links the stop usages of the two journey patterns. The vehicle journey interchange links the two vehicle journeys.

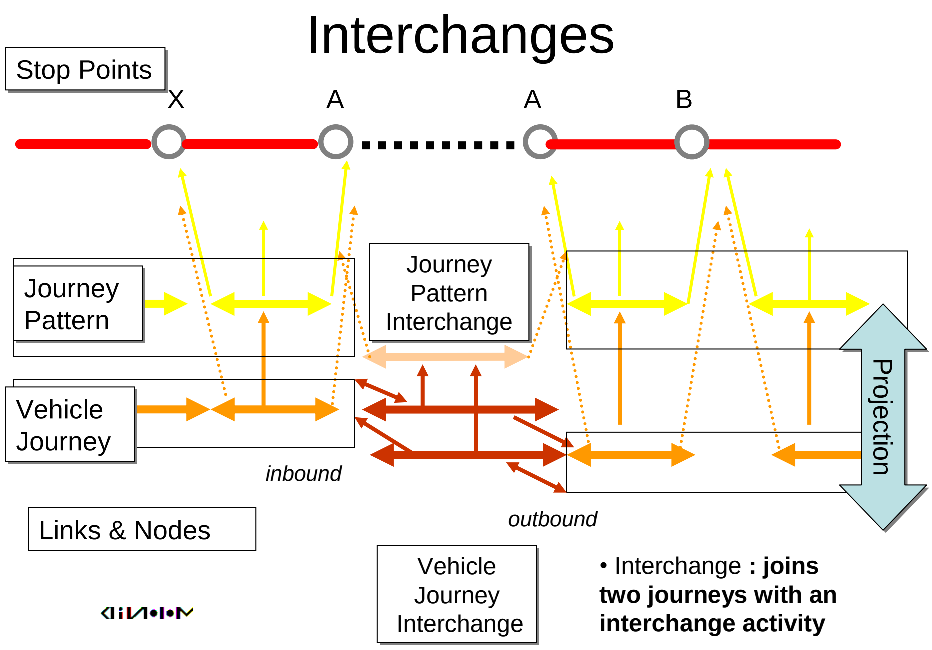


Figure 3-41 – Interchange Links

3.11.3 Interchange Instance Example

As a pictorial example of a connection, *Figure 3-42* shows a UML instance diagram of the element instances for a connection between two vehicle journeys:

- At the top, in yellow, can be seen a ***Service*** with two journey patterns, one inbound feeder to a ***JourneyPatternInterchange,*** and one outbound distributor from it. Each ***JourneyPattern*** has a single ***JourneyPatternSection*** containing a sequence of timing links; only the last ***JourneyPatternTimingLink*** of the inbound feeder journey pattern and the first ***JourneyPatternTimingLink*** of the outbound distributor journey pattern are shown, along with the ***VehicleJourneyStopUsage*** instance for each end of the link.

- The ***JourneyPatternInterchange*** instance references both inbound feeder and outbound distributor journey patterns. It also references the destination ***VehicleJourneyStopUsage*** instance of the last timing link of the inbound feeder pattern, and the origin ***VehicleJourneyStopUsage*** of the last timing link of the outbound distributor pattern.

- Below this, in orange, can be seen two corresponding inbound feeder and outbound distributor ***VehicleJourney*** instances. Again, only the last ***VehicleJourneyTimingLink*** of the inbound feeder vehicle journey and the first ***VehicleJourneyTimingLink*** of the outbound distributor vehicle journey are shown.

- .Each ***VehicleJourneyTimingLink*** individually projects onto the appropriate ***JourneyPatternTimingLink*** instance by an explicit reference.

- Each vehicle journey has its own instance of a ***VehicleJourneyInterchange,*** which references both the inbound feeder and outbound distributor vehicle journey instances. It also references the ***JourneyPatternInterchange*** that connects the journey patterns upon which the vehicle journeys are based.

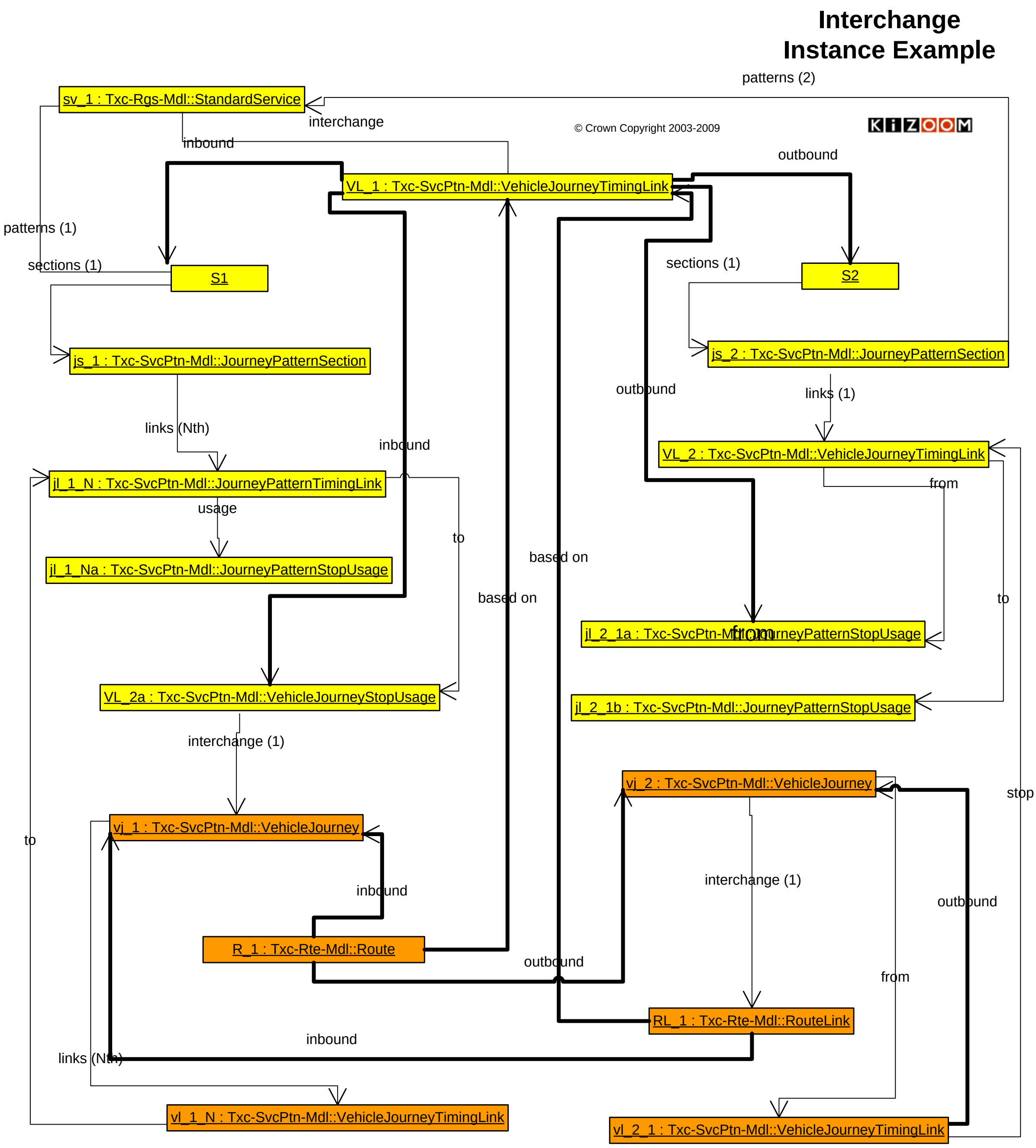


Figure 3-42 – UML Instance Diagram of Example Interchange

3.12 Fare Stages

TransXChange supports the annotation of links with basic fare stage data for operational purposes. There are two different models commonly used for fare stages:

1. A **Stage Fare** model, where the fare stage is located on a boundary between two zones and is considered to be in both zones. In effect the fare stage is on the stop point, but only applies to journeys (i.e. sequences of links) where the other end of two subsequent links is in different zones.
2. A **Zonal** model, where the fare stage boundary lies between two stops, each within a distinct fare zone (***TariffZone***). The fare stage is in effect on the link between the stops. Only journeys going in the direction of the other zone and that cross the boundary will encounter the fare stage. One or more ***TariffZones*** can be specified for a stop using NaPTAN 2.5

In the *TransXChange* model, fare stages are a property of timing link stop usage, so that both Stage Fare and Zonal models can be supported. Fare stage values can be specified at both the journey pattern and vehicle journey level of discourse as a successive property, that is one that carries onto succeeding links in the series until reset.
The fare stage change occurs at the point of pick up, that is, at the originating end of the link, as shown in *Figure 3-43,* which shows examples of link sequences over a zone boundary for both fare models, with fare stage numbers and fare stage points marked. Whether a stop usage for a given link is a fare stage is properly determined by whether the ***FareStageNumber*** changes when traversing a sequence of timing links: the ***FareStage*** indicator can be used to store a statically computed determination of this property for convenience of implementation.

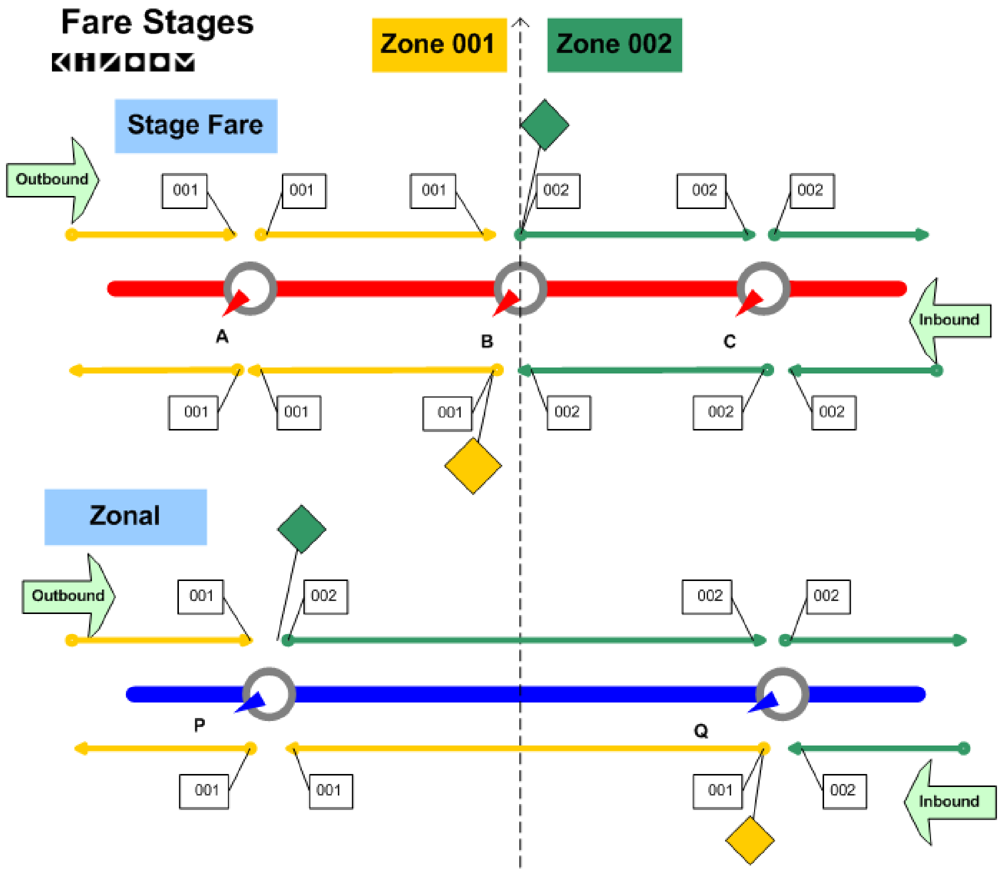


Figure 3-43 – Fare Stages & Links

3.13 Dead Runs

‘Dead run’ or positioning runs may be specified on vehicle journeys to describe how vehicles are placed in position to deliver a service, and also how they are retrieved after completing the service. Dead run positioning links are primarily of use for exchanging information for AVL systems, and are not needed for registration or publishing schedules. Dead runs can also be used to indicate short working. *Figure 3-44* shows a UML structure diagram of the elements used to describe dead runs.

- A ***VehicleJourney*** may have an initial ***StartDeadRun*** and a final ***EndDeadRun***.
- Each ***DeadRun*** consists of one or more ***PositioningLink*** instances.
  - Each ***PositioningLink*** runs between two position points, which may be specified as either a ***Location***, a ***StopPoint***, a ***LayoverPoint*** or a ***Garage***.
  - A ***DeadRun*** may reference a ***VehicleJourneyTimingLink*** to indicate the point at which short working starts or stops.

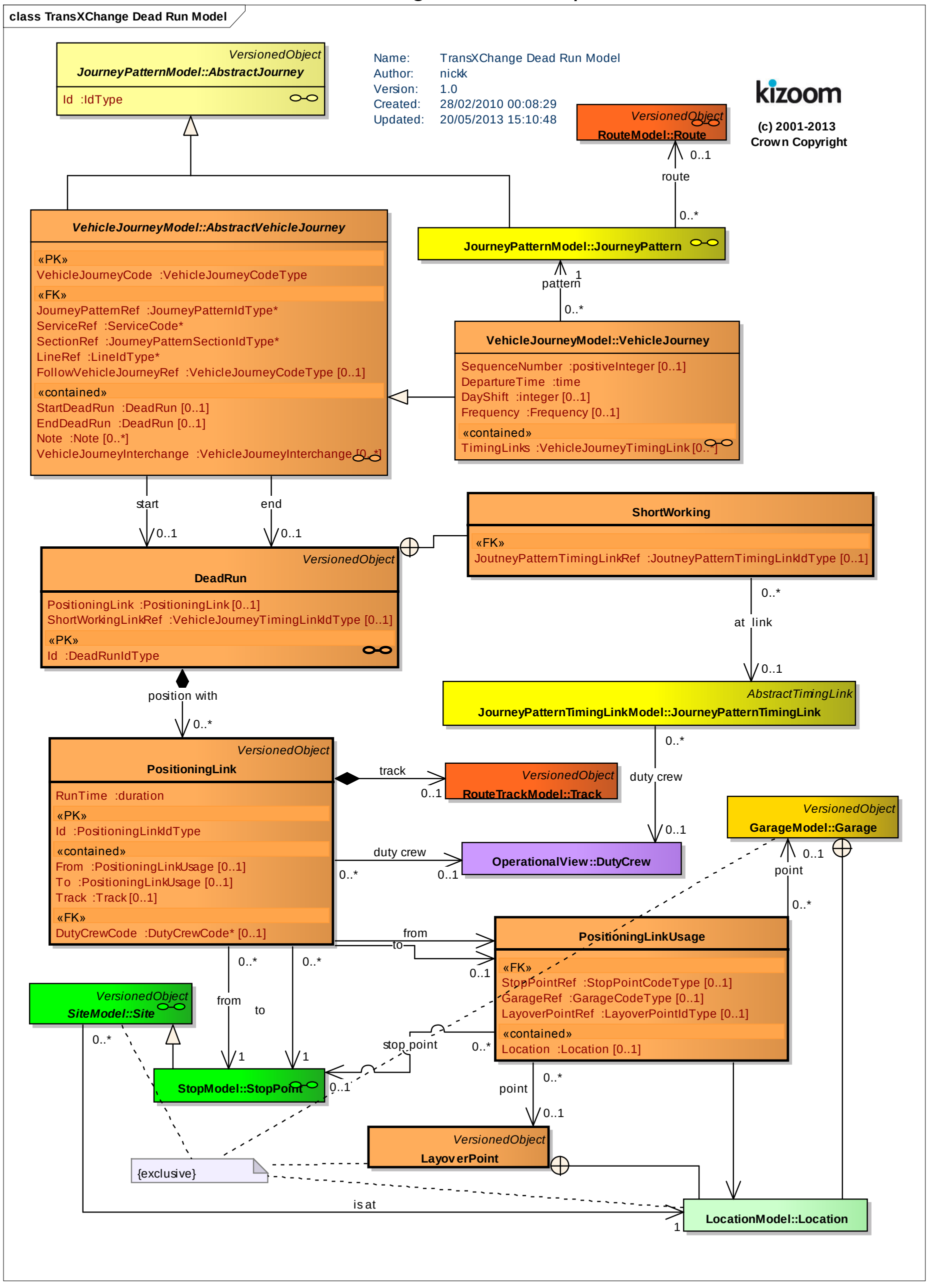


Figure 3-44 – UML Diagram of Dead Run Model

3.13.1 Use of Dead Runs for Short Working

Dead runs may be used to indicate that a Vehicle Journey starts or ends at a particular point in a journey pattern, omitting all links & stops before or after the intercept point. See the *Circular Route* example for an illustration of both short and full workings of the same route.

3.13.2 Dead Run Hierarchy

Figure 3-45shows the inheritance hierarchy for the ***DeadRun*** model elements.

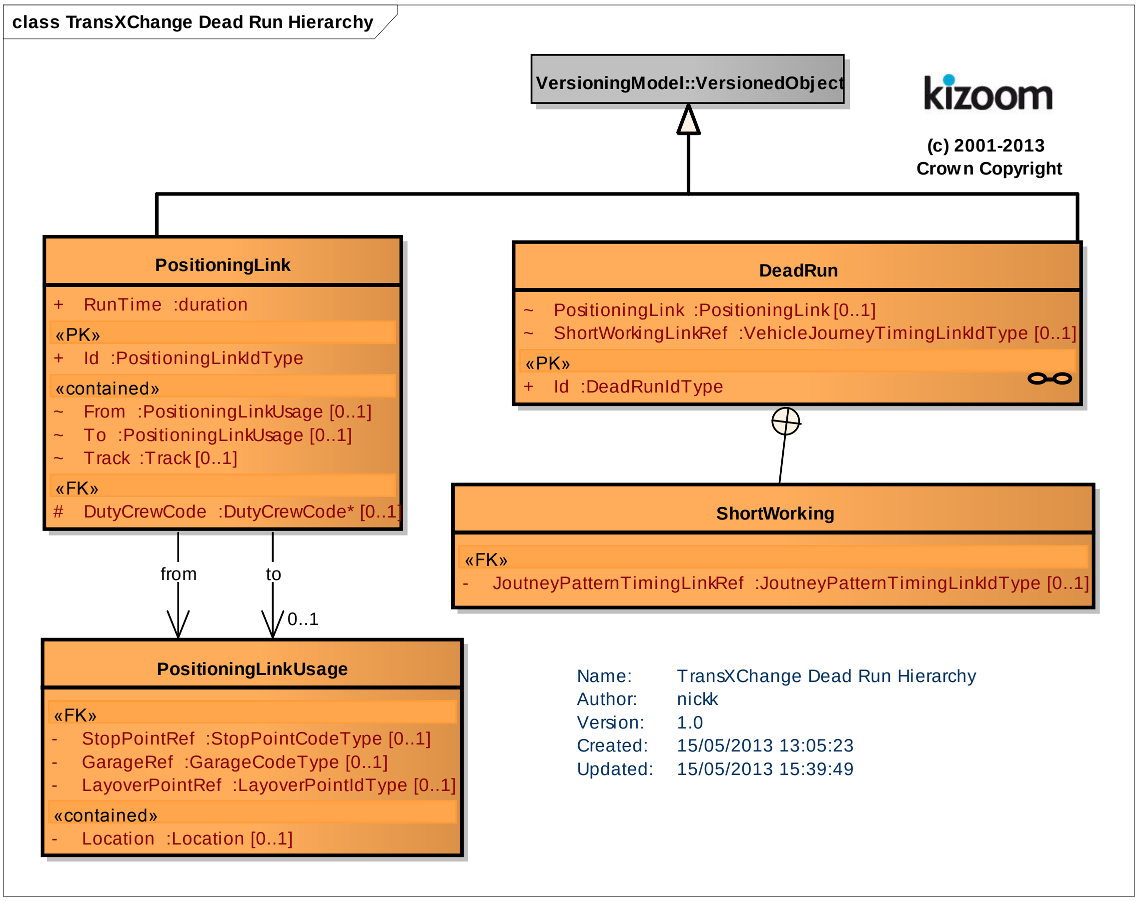


Figure 3-45 – UML Diagram of Dead Run Model Element Hierarchy

3.14 The Registration Model

The statutory requirements of a bus registration are captured in *TransXChange Registration* by a submodel of ***Registration*** descriptive elements.

3.14.1 Introduction to the Registration Model

Figure 3-46 introduces the elements used to represent a ***Registration***. These comprise:

- A TransXChange document can contain a ***Registration*** element:
  - A *TransXChange Registration* document must contain one ***Registration*** instance.
  - A *TransXChange General* document may contain one or more ***Registration*** instances.

- A single ***Service*** can be associated with each ***Registration***.
  - A *TransXChange Registration* document ***Registration*** must contain a ***Service*** instance that references the ***Registration***. It may have other ***Service*** definitions for connecting services.
  - A *TransXChange General* document may contain a ***Service*** instance.

- A ***Service*** has a ***RegisteredOperator,*** and may have additional ***AssociatedOperator*** instances. Operators may be instances of either ***LicensedOperator*** or ***Operator.***
  - In a *TransXChange Registration,* the ***RegisteredOperator*** must be a ***LicensedOperator*** instance, with all details completed. (Note this constraint is enforced by an XML keyref).
  - In a *TransXChange General* document the ***RegisteredOperator*** may be an instance of either ***LicensedOperator*** or ***Operator.***

- A ***Registration*** records the ***TrafficAreaNetwork*** and ***CirculatedAuthority*** instances.
  - Additional special details can be recorded for a ***ShortNoticeRegistration,*** including references to other services that the service replaces, or to which it connects. A short notice registration is an application to register, cancel or change a service made with less than the normally required 56 days' period of notice.
  - The ***Registration*** can be annotated with ***SupportingDocument*** instances that identify related documents.

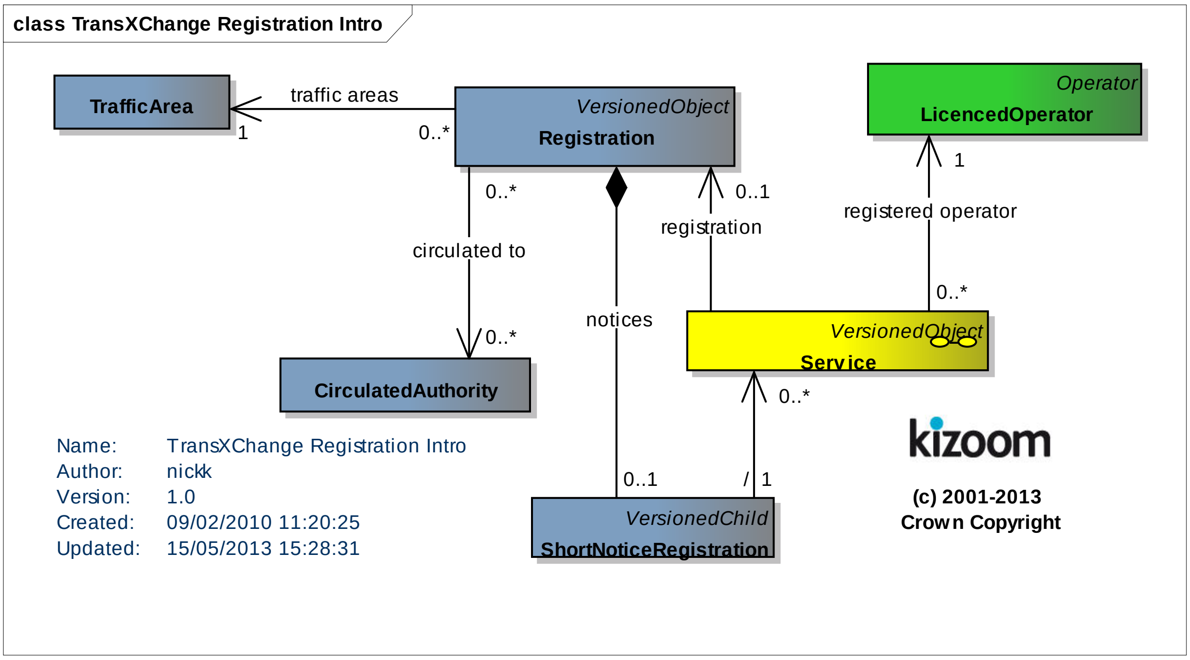


Figure 3-46 – UML Diagram of Basic Registration Model

3.14.2 Registration Model Details

Figure 3-47 elaborates *Figure 3-46* to show the attributes of ***Registration*** elements.

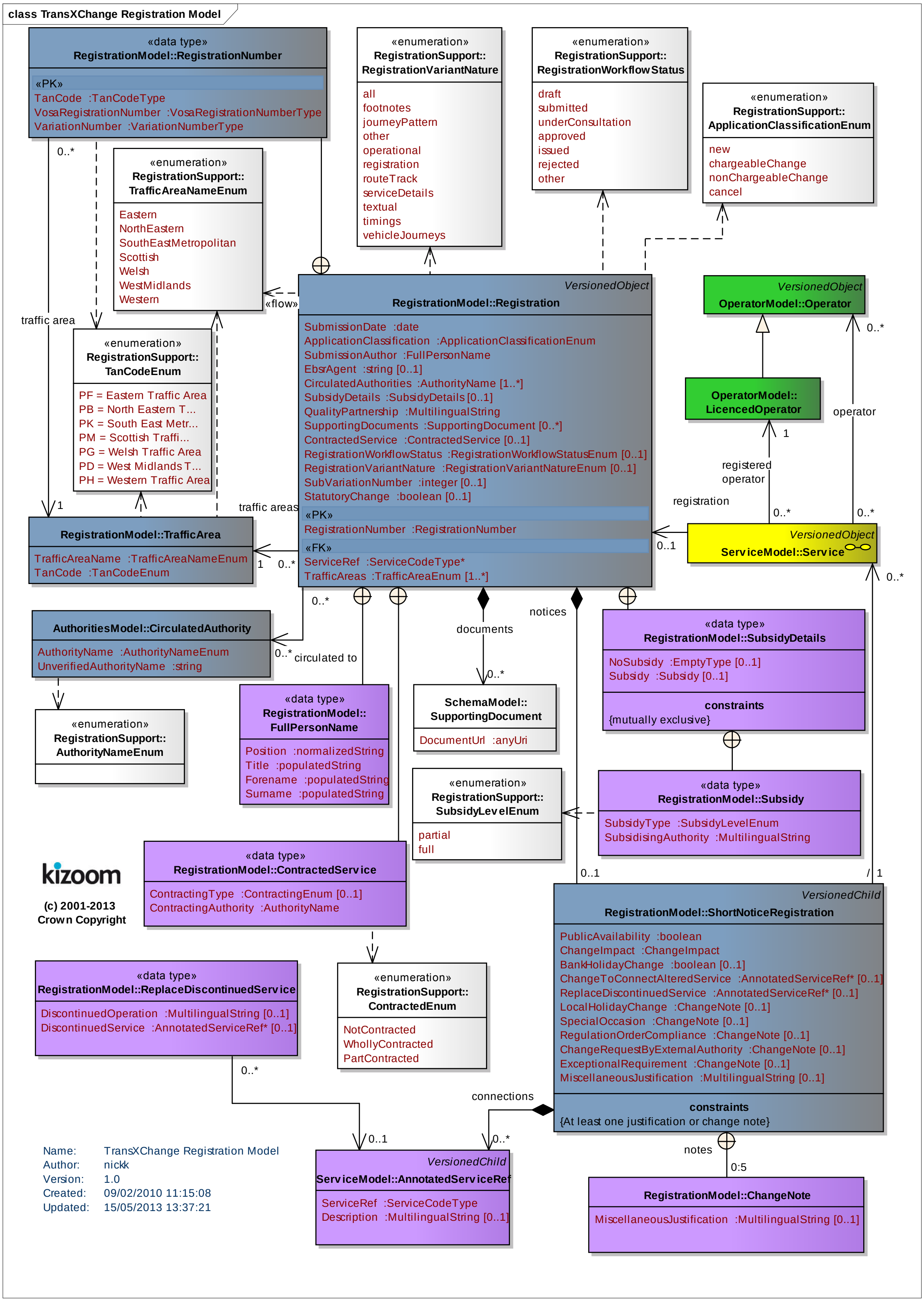


Figure 3-47 – UML Diagram of TransXChange Registration

3.14.3 Registration Model Workflow States

The ***WorkflowStatus*** attribute allows the status of a ***Registration*** to be represented for automated workflow processes. Figure 3-48 shows the allowed states and transitions.

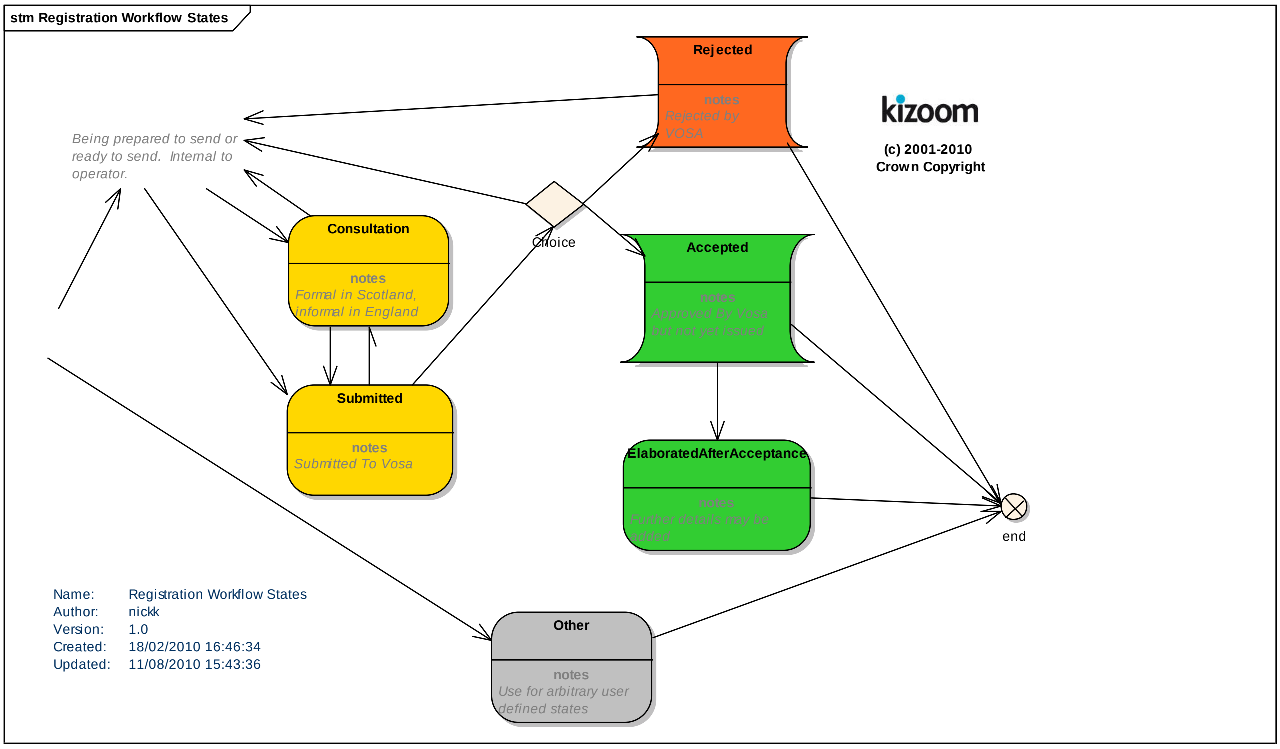


Figure 3-48 – State transitions for registration Workflow Status

3.14.4 Registration Model Hierarchy

Figure 3-49 shows the inheritance hierarchy for the Registration model elements.

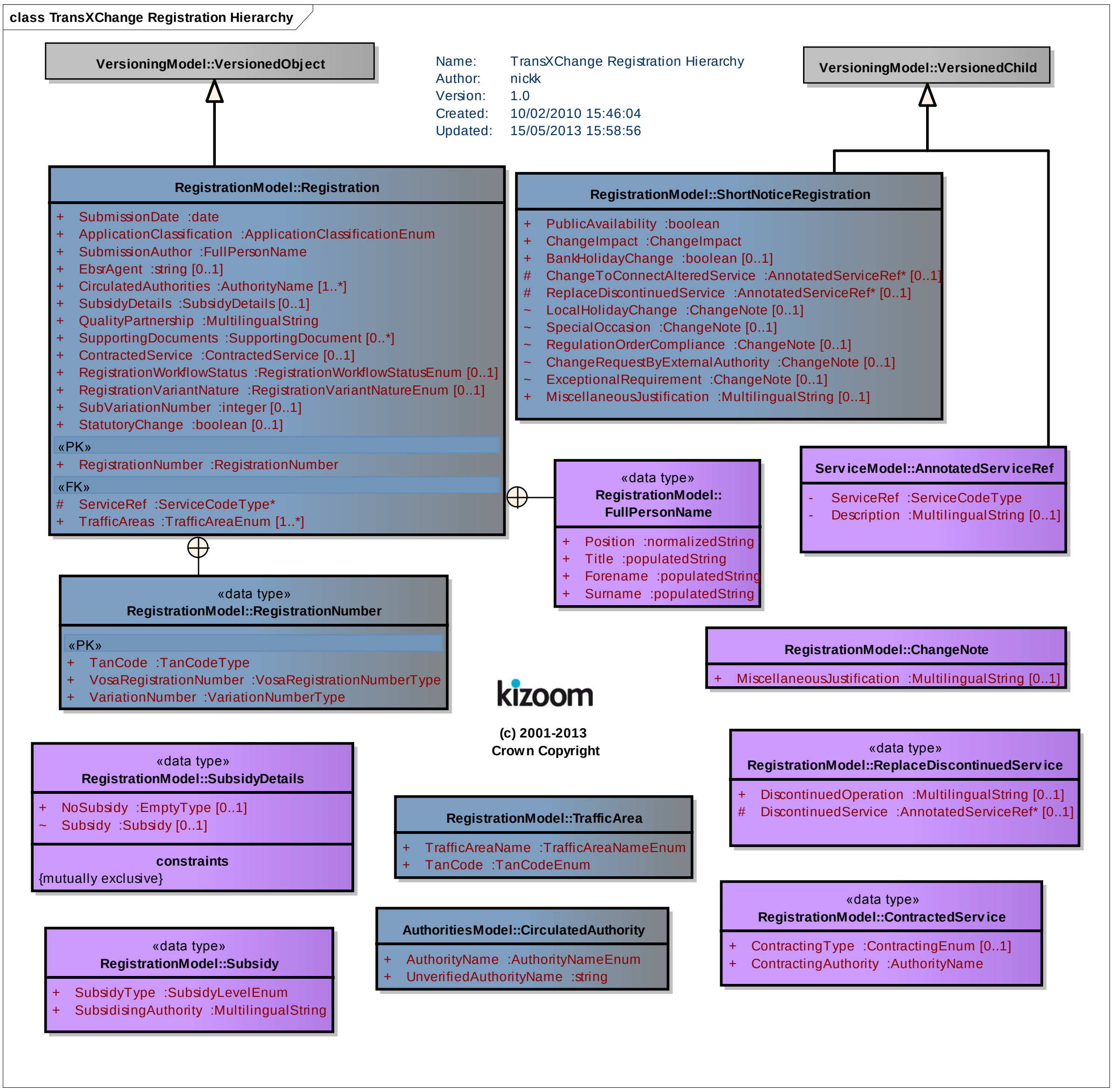


Figure 3-49 – UML Diagram of Registration Element Hierarchy

3.14.5 Registration Model Support Types

Figure 3-55 shows the additional data types used the ***Registration*** model elements.

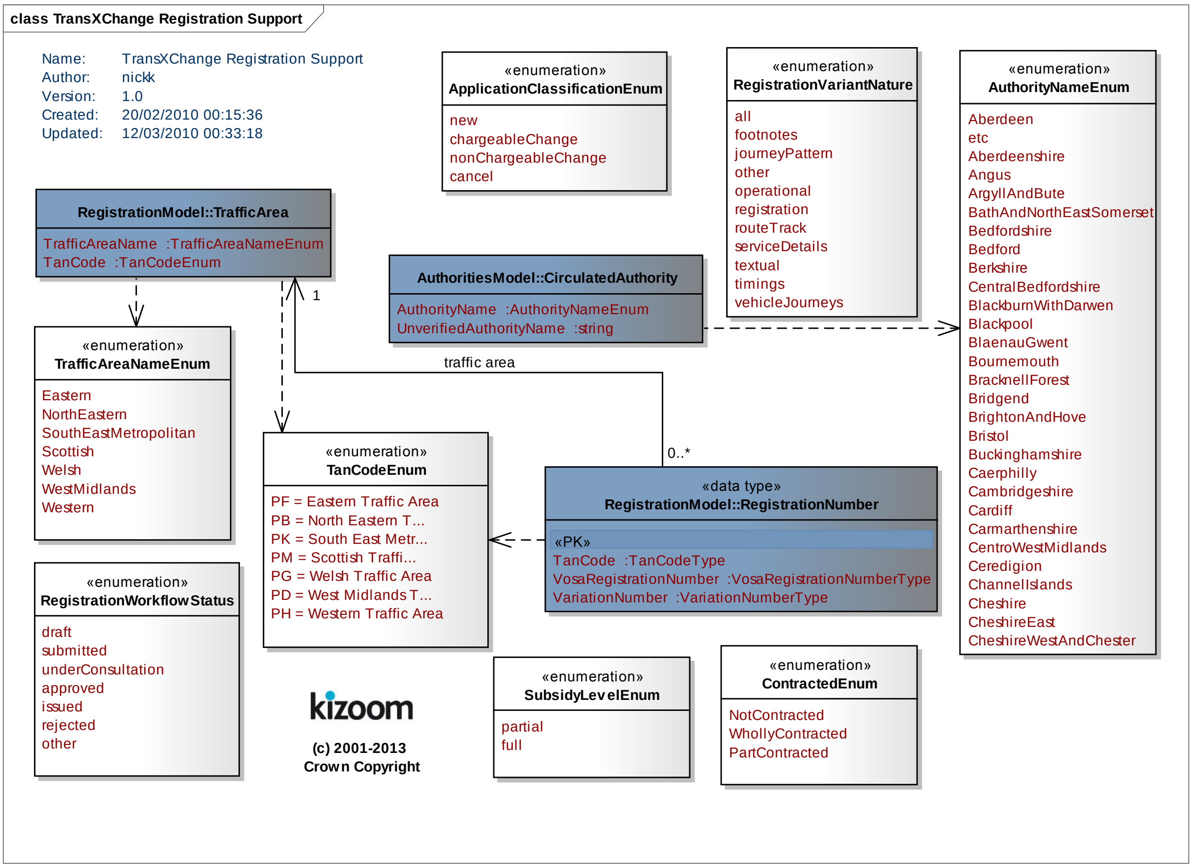


Figure 3-50 – UML Diagram of Registration Support types

3.14.6 Populating a Registration

Although it is legitimate for a Single Registered Service to have a number of journey pattern variants, the variation should be less than 50% of the primary journey pattern; i.e. *more than 50% of the mileage of the journeys should be in common, i.e. consist of vehicle journeys with timing links that visit or pass the same stops in the same order.*

3.14.7 Cancellation a Registration

When cancelling a registration it is not necessary to provide the entire schedule in electronic format. A restricted set of elements can be provided sufficient to identify the Registration and key particulars - the ***Registration***, the ***Service*** & the ***Operator***. AN example is provided.

3.14.8 Services and Registrations

Each registration file should contain all the journeys for a given service. Journeys which are operated by the same operator, use the same line number and follow the same journey pattern within the same operating period, should be in the same registration.

3.15 Operators

3.15.1 Introduction to the Operator Model

TransXChange includes a basic representation of an ***Operator*** to record who is making a Registration and who provides a Vehicle Journey. *Figure 3-51* introduces the elements used to represent an ***Operator***.

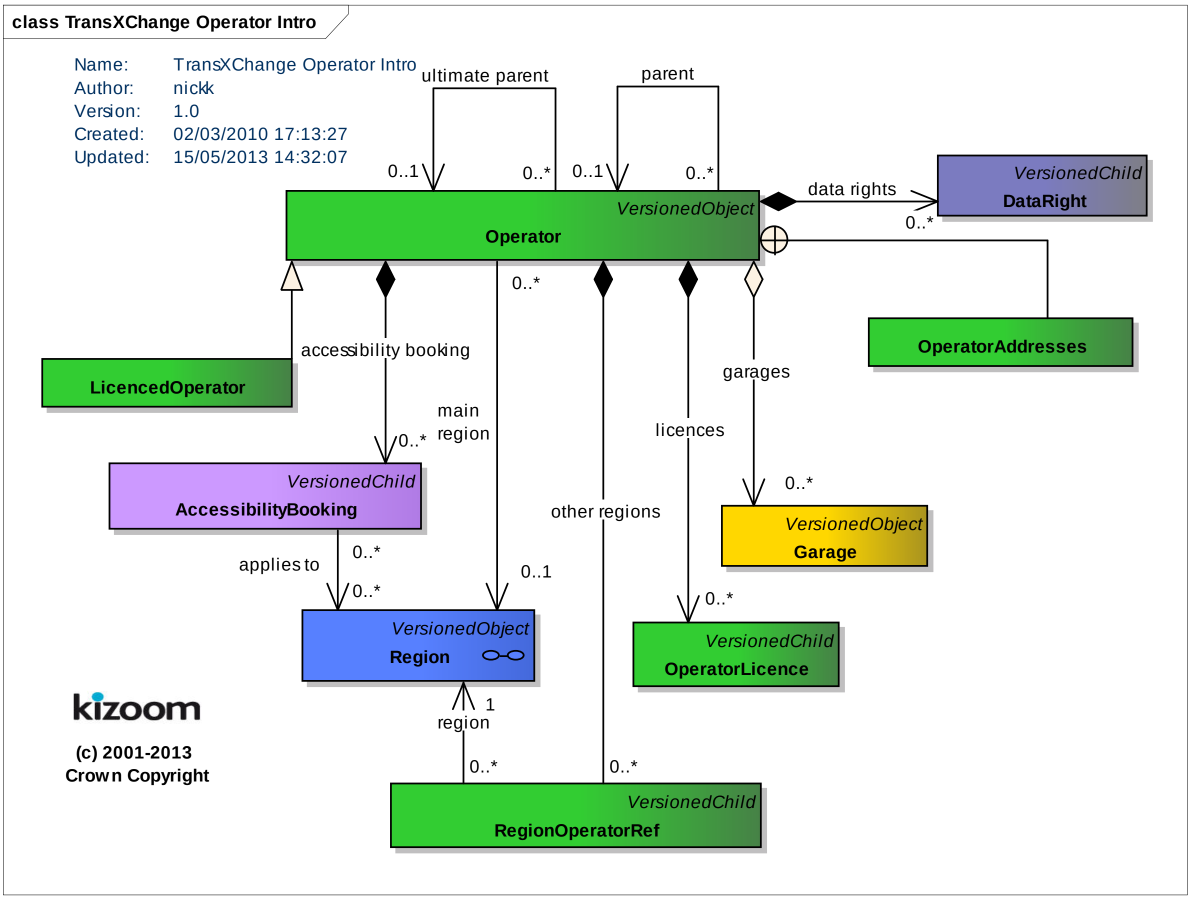
- ***LicensedOperator*** is a specialisation of ***Operator*** adds details about the licence or licences of the operator.
- An operator may have associated **Garages** and also **DataRights** (See later).
- ***AccessibilityBooking*** (V2.5) holds contact details for wheelchair bookings for the operator's service. (See later).
- ***RegionOperatorRef*** provides a mapping of the Operator's code for different regions if needed.
- 

Figure 3-51 – UML Model of Operator

3.15.2 Operator Model Details

Figure 3-52 elaborates *Figure 3-51* to show the attributes of ***Operator*** elements.

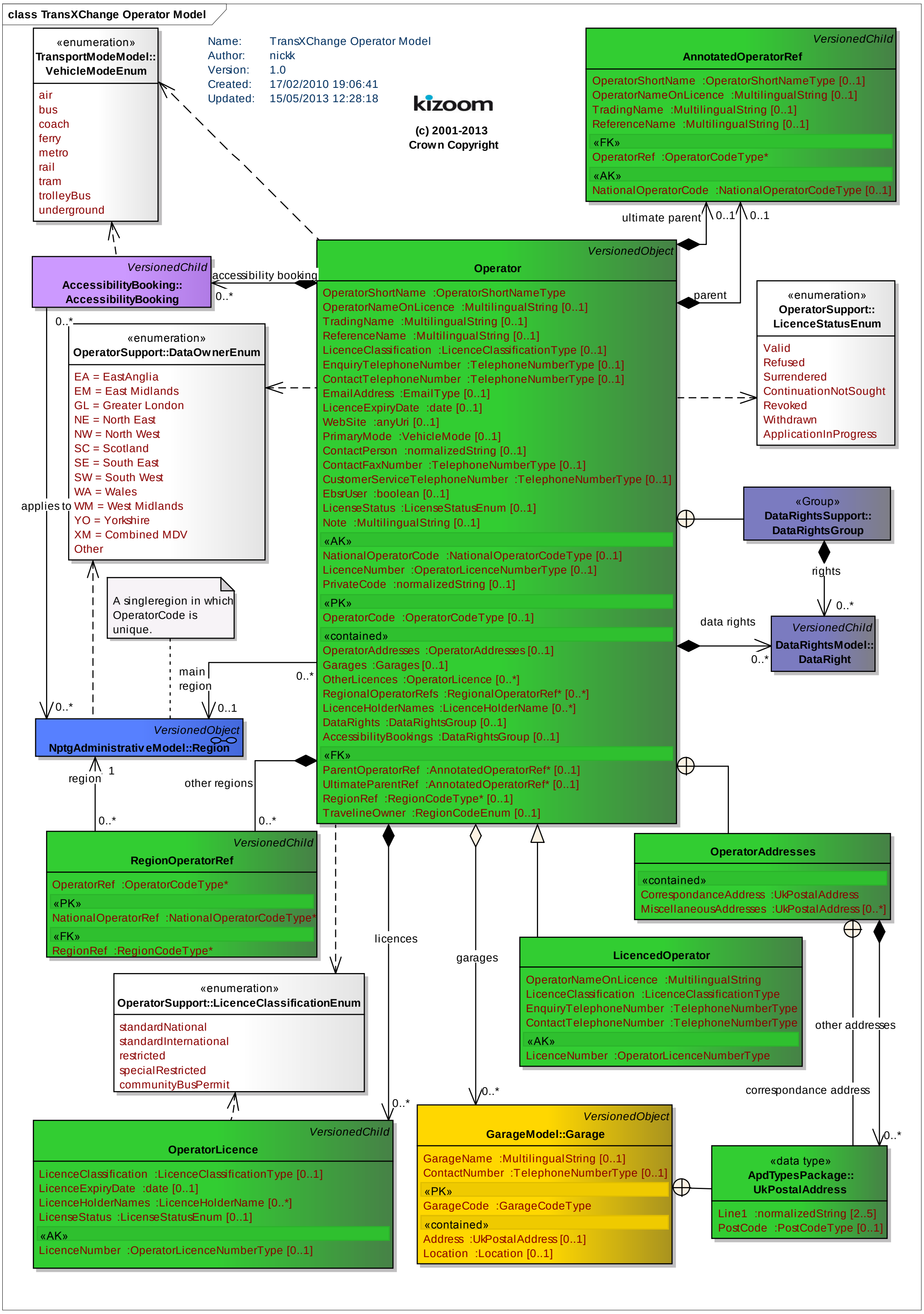


Figure 3-52 – UML Diagram of TransXChange Operator Model

3.15.3 Operator / Accessibility Booking ((v2.5)

AccessibilityBooking *(*Figure 3-53*)* allows details about booking for assistance for wheelchair users to be recorded for an operator. Different values may be specified for different modes and regions.

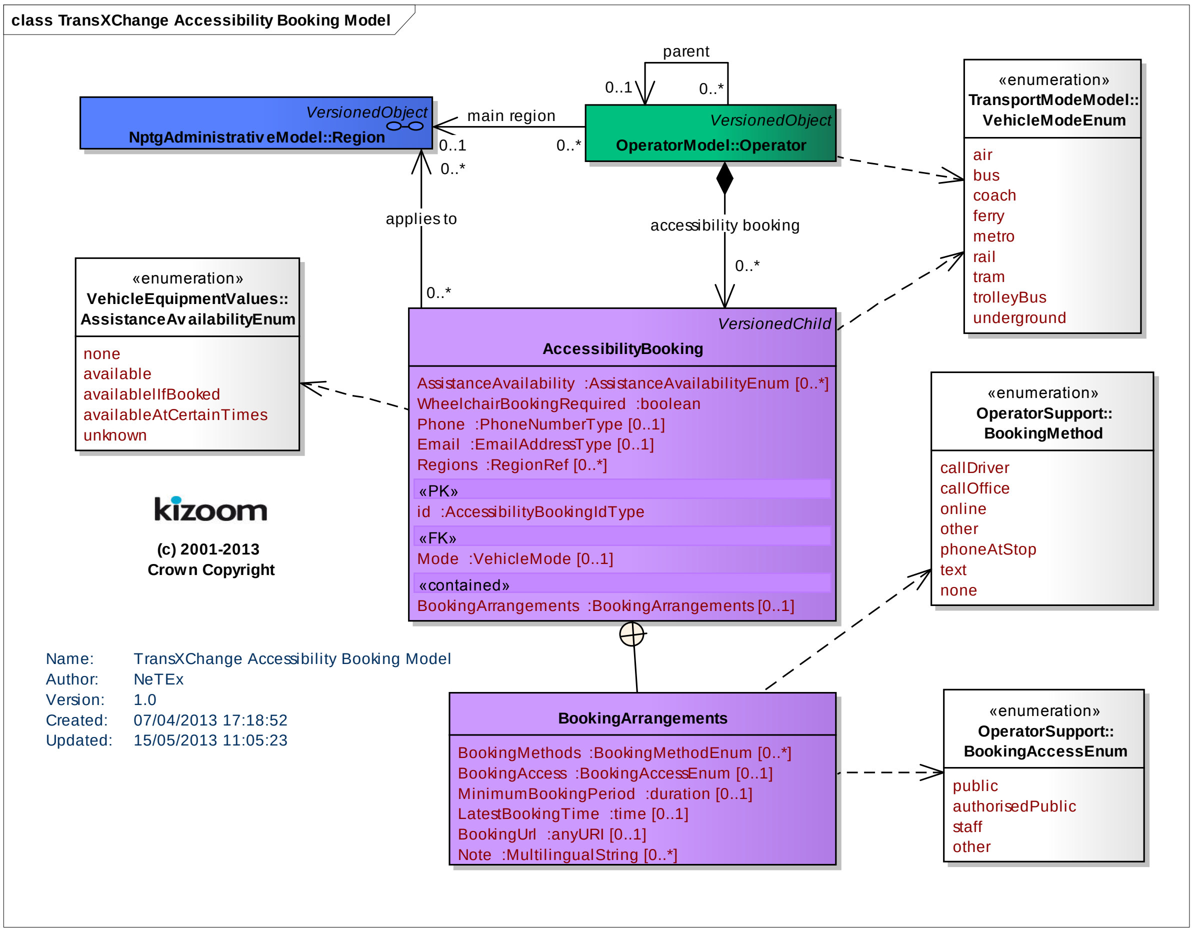


Figure 3-53 – UML Diagram of TransXChange Accessibility Booking Model

3.15.4 Operator Model Hierarchy

Figure 3-54 shows the inheritance hierarchy for the ***Operator*** model elements.

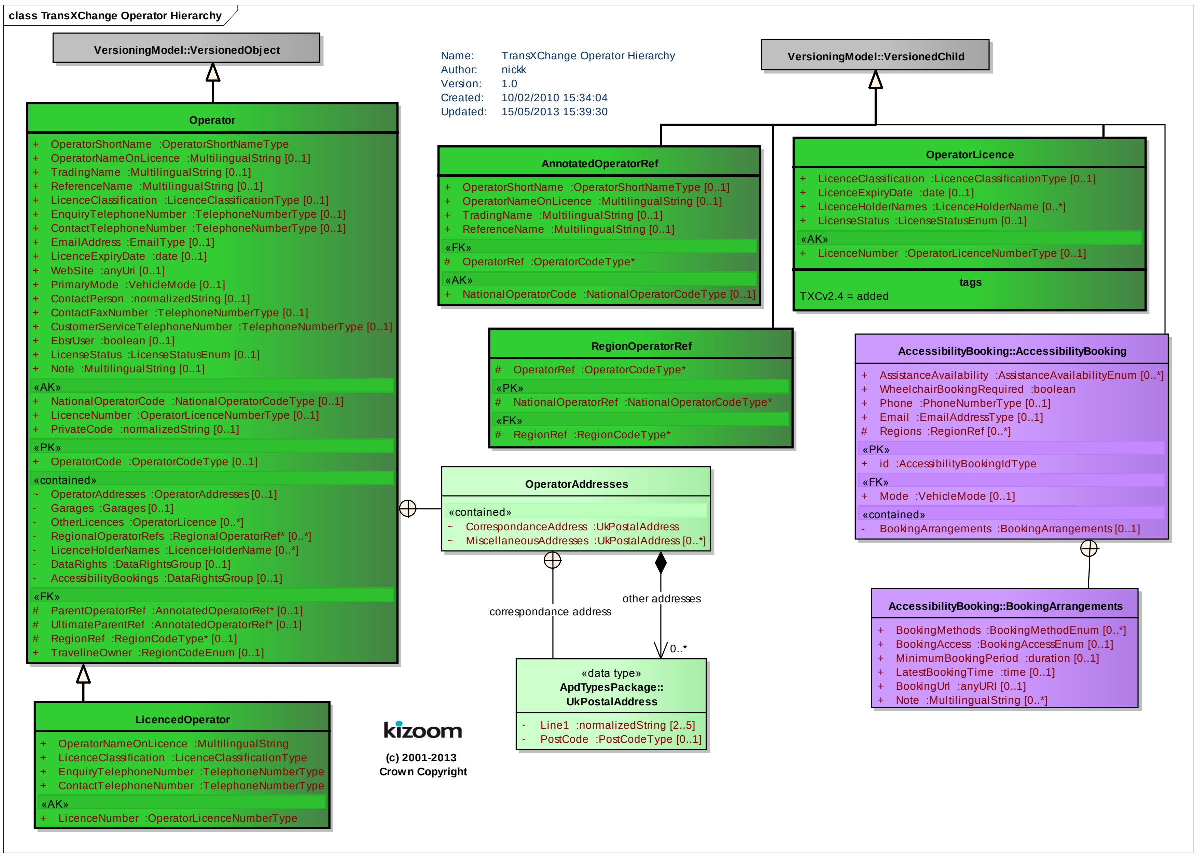


Figure 3-54 – UML Diagram of Operator Element Hierarchy

3.15.5 Operator Model Support Types

Figure 3-55 shows the additional data types used the Operator model elements.

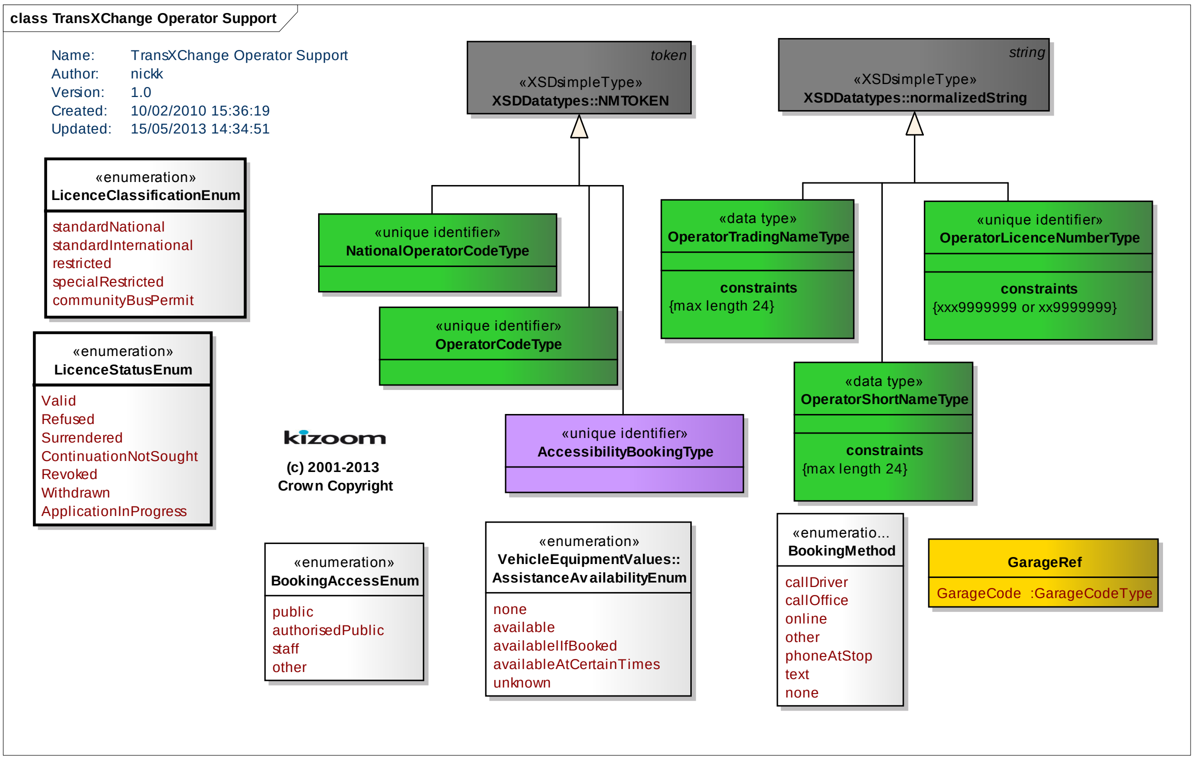


Figure 3-55 – UML Diagram of Operator Support types

3.16 Further Modelling Topics

3.16.1 Direction: Handling Inbound and Outbound Schedules.

A Service may contain both inbound and outbound journeys, comprising in effect two distinct timetables for the two directions. Normally completely separate routes will be specified for each direction, because there are typically separate NaPTAN points for bus stop pairs each side of the road; routes will therefore be following a different sequence of stops along slightly different road sections. However, there are scenarios where the route (and associated sequence of stops) in one direction is an exact reversal of the route (and associated sequence of stops) in the opposite direction. In this case it is possible to share the route definitions for both directions of a service, as follows (*Figure 3-56*).

1. Each Route contains one or more route sections, each containing a sequence of route links. Each route link is flagged as *Outbound, Inbound, Clockwise* or *Anticlockwise*. All the links within a route section must be in the same direction.

2. At least one journey pattern is specified for each direction of the Route. The journey pattern sections contain journey pattern timing links in the order of traversal, each of which can specify a direction (if a direction is not specified the direction will be assumed to be the same as that of any route link which the timing link references).
   - If the direction of a journey pattern timing link is the **same** as that of the route link which it references, then the stops referenced in the "*from"* and "*to*" stop usages of the timing link will be the same as for the route link, and the timing links will appear in the same order as the route links.

   - For example, if (a) Route Link '*RL_1'* goes from '*A'* to '*B'* with a direction of '*outbound'*, and (b) Route Link '*RL_2'* goes from '*B'* to '*C',* also with a direction of '*outbound'*, then the outbound journey pattern would have two outbound journey pattern timing links: (i) '*JTL_1'* which references '*RL_1'* with a direction of '*outbound'*, and also runs from '*A'* to '*B'*, followed by (ii) journey pattern timing link '*JTL_2'*, which references '*RL_2, and* goes from '*B'* to '*C'.'* Note that in this discussion; 'A', 'B', etc refer to stop pairs: in actuality, the inbound and outbound stops are likely to be distinct stops of a pair either side of the road. So actually the NaPTAN stops of inbound and outbound routes and journey pattern will be quite distinct.
- If the direction of the of a journey pattern timing link is the **opposite** to that of the route link which it references, then both the link order, and the stops referenced in the from and to stop usages will be reversed.
   - For example, if (a) Route Link '*RL_1'* goes from '*A'* to '*B'* with a direction of '*outbound'*, and (b) Route Link '*RL_2'* goes from '*B'* to '*C',* also with a direction of '*outbound'*, then the inbound journey pattern would have two inbound journey pattern timing links: (i) journey pattern timing link '*JTL_X1,'* which references '*RL_2'* but which runs from 'C' to 'B', and (ii) journey pattern timing link '*JTL_X2,'* which references '*RL_1'* but which runs from 'B' to 'A'.

3. Each vehicle journey follows the same direction as the journey pattern that it references.

4. The Service may be given an overall Direction: this may be one of *Inbound, Outbound, InboundAndOutbound, Clockwise, Anticlockwise*, or *Circular*.

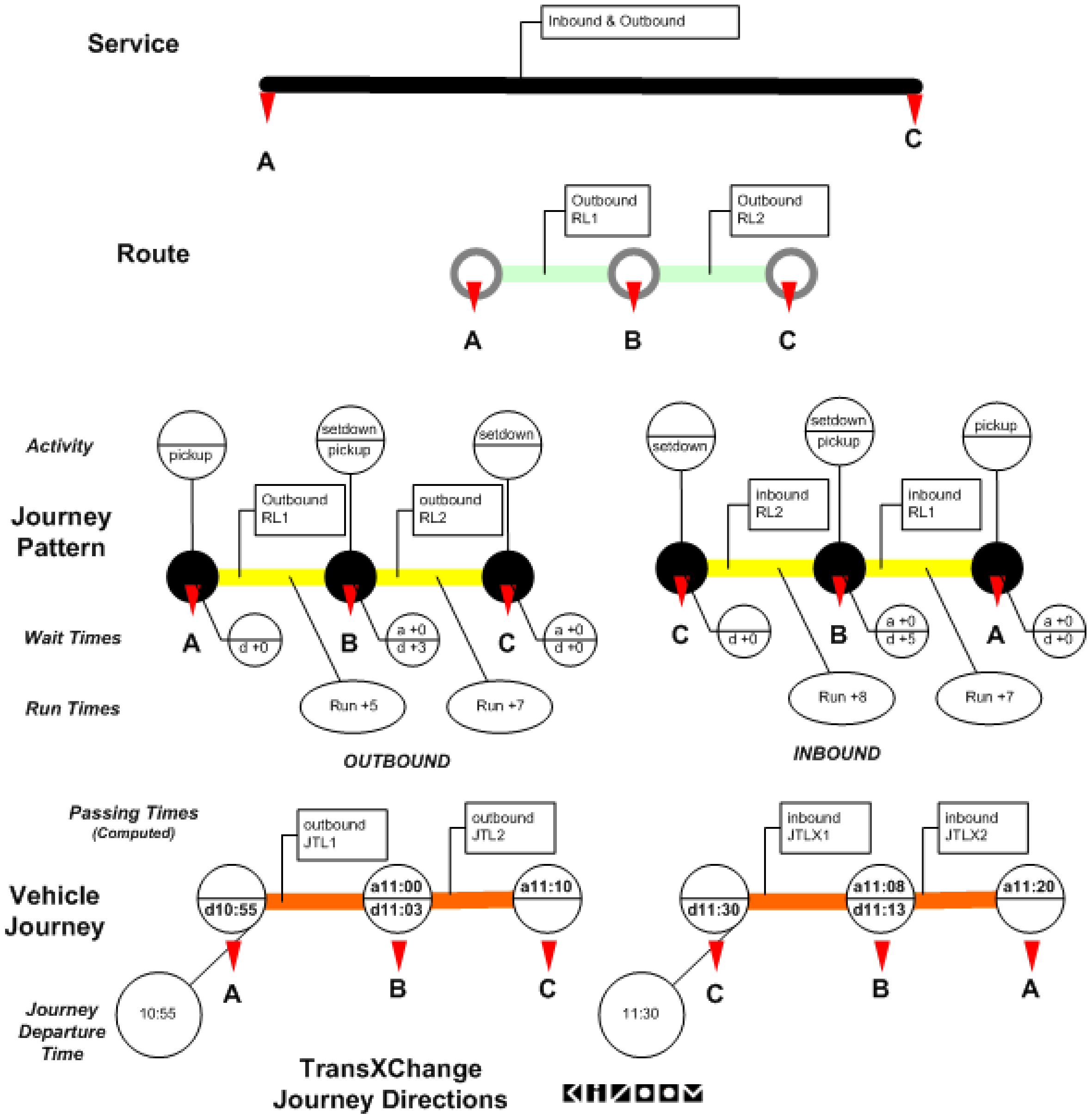


Figure 3-56 – Journey Directions

The *TransXChange Publisher* will sort the vehicle journeys of a service into distinct outbound and inbound groups, and create a separate matrix for each direction.

3.16.2 Modelling Complex Routes

The *TransXChange* model can be used to represent complex services, for example:

- Services with topologically complex routes.
- Services with complex temporal operational patterns.

3.16.2.1 Services with Topologically Complex Routes

The *TransXChange* model can be used to represent complex patterns of service:

1. **Repeated stop routes.** *Circular* (*Figure 3-57*), *Lollipop* (*Figure 3-58*) and *Cloverleaf* (*Figure 3-59*) routes involve visiting the same stop more than once within a single vehicle journey. In the *TransXChange* model, each link has a separate identity in both the route, journey pattern and vehicle journey link sequences, so it is possible to distinguish the separate link traversals and occurrences of a stop in a journey, and so to compose complex routes, and also to

project unambiguously the links of such routes between the route, journey pattern and vehicle journey level of discourse. (In *TransXChange* 1.2 this was not always possible). Other features helpful in representing complex routes are:

- **Dynamic destination displays**, so that bus headings can change over the course of the route.
- Reusable route and journey pattern **sections**, so that definitions of sections of the route and/or journey pattern may be shared between different journeys. See 'Modelling Services Efficiently' below.
- Stop **Sequence numbers** – so that the presentation of a route in a matrix can be exactly controlled. See 'Presenting Schedules in Timetables' below.

2. **Multiple route variants**. Complex services may be composed of multiple route and journey pattern variations, either involving covering different branches of the physical network, or traversing subsets of the full stop sequence, or both.
    - ***Line*** elements can be used to separate the *modelling* of the network topology as routes and journey patterns, from the *labelling* of the network services with public identifiers on vehicle journeys – which is done using the ***Line*** / ***LineName*** element. Thus several different route variants may all be grouped under the same line name.
    - ***RouteSection*** elements can be used to model reusable subsections and branches of the route network, and ***JourneyPatternSection*** elements can be used to annotate this substructure with timing values, allowing for the representation and reuse of the route substructure.

3. **Connecting routes**. The connections between routes services may be described using ***JourneyPatternInterchange*** and ***VehicleJourneyInterchange*** elements.

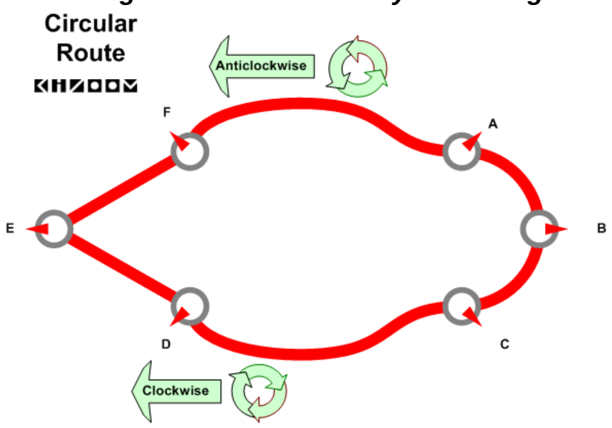


Figure 3-57 – Topology: Circular Route

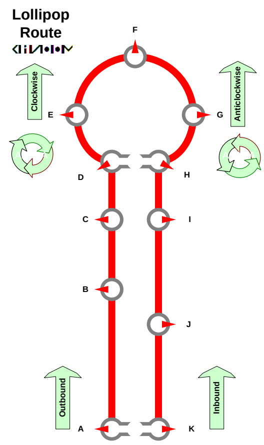


Figure 3-58 – Topology: Lollipop Route

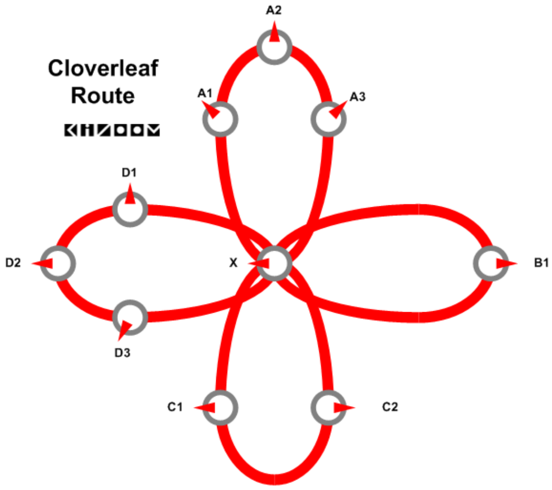


Figure 3-59 – Topology: Cloverleaf Route

3.16.2.2 Services with Complex Temporal Operational Patterns

The *TransXChange* model can be used to represent complex operational times. This is discussed in detail under ‘Modelling Operational Days’ below. All of the following mechanisms are available:

- **Regular day types:** Days of week, Day Combinations, Weeks of month.
- **Special day types:** Bank holidays.
- **Date ranges:** Operating Period, Validity Periods, and Exceptions.
- **Time bands:** Time bands of operation of flexible services.
- **Frequency based:** Interval or minute patterns of operation of frequency based services.
- **Serviced Organisations:** Term-times, holidays or other events of specified organisations.

3.16.3 Modelling Services Efficiently

TransXChange supports an extensive reuse of service and journey description elements so that an efficient encoding of journeys can be achieved. In particular:

- Existing data reference systems can be used; *NaPTAN* stop and stop area definitions, *NPTG* localities.
- Elements describing the network topology and other shared infrastructure entities can be declared once, and then be reused by a simple reference. Notably:
    - Topographical elements: ***StopPoint***, ***StopArea***.
    - Organisational elements: ***ServicedOrganisation, Operator***.
    - Network layer elements: ***Route, RouteSection***.
    - Supply elements: ***JourneyPatternSection,*** and ***JourneyPattern***.
- Link Sequences – the timing link sequences of journey patterns and vehicle journeys – may be reused in several different ways:
    a. A ***JourneyPatternSection*** may be used in many different ***JourneyPattern*** instances.
    b. A given ***JourneyPattern*** may be referenced in many different ***VehicleJourney*** instances, and its values inherited. Only those individual vehicle journey timing links whose properties are different from the corresponding timing links of the underlying journey pattern need be specified.
    c. A ***VehicleJourney*** may specify that particular stops of a referenced ***JourneyPattern*** are omitted, allowing for "express" journeys constrained to a basic journey, and for short working.
    d. A ***VehicleJourney*** may reference another ***VehicleJourney*** to share the timing links of that specific journey.
    e. The ***Frequency*** element of ***VehicleJourney*** may be used to indicate that the same vehicle journey is repeated to the same pattern many times at regular intervals.
- Operational day types and dates may be reused.
    - The ***OperatingProfile*** specified for a ***Service*** can be shared by all the service's vehicle journeys. Individual ***JourneyPattern*** and ***VehicleJourney*** instances need only state their specific differences from the base values.
- Properties of successive links need only be specified when they change:
    - The successive properties of links, such as fare stages and dynamic headings, do not have to be repeated on every link, but only need to be specified when they change from the preceding link.

It remains up to the implementer to decide the degree of reuse that she wishes to achieve. A verbose implementation may, if it wishes, re-declare stops and create separate route, route section, route links, journey pattern, journey pattern section, journey pattern links, operation profile, and special operation profile instances for every single vehicle journey. However it should be note that a verbose implementation (a) wastes space (b) may fail to exchange structural information about the underlying schedule.

3.16.3.1 Overall Reuse of Elements

Figure 3-60 shows some of the ways that elements may be reused at different levels of discourse.

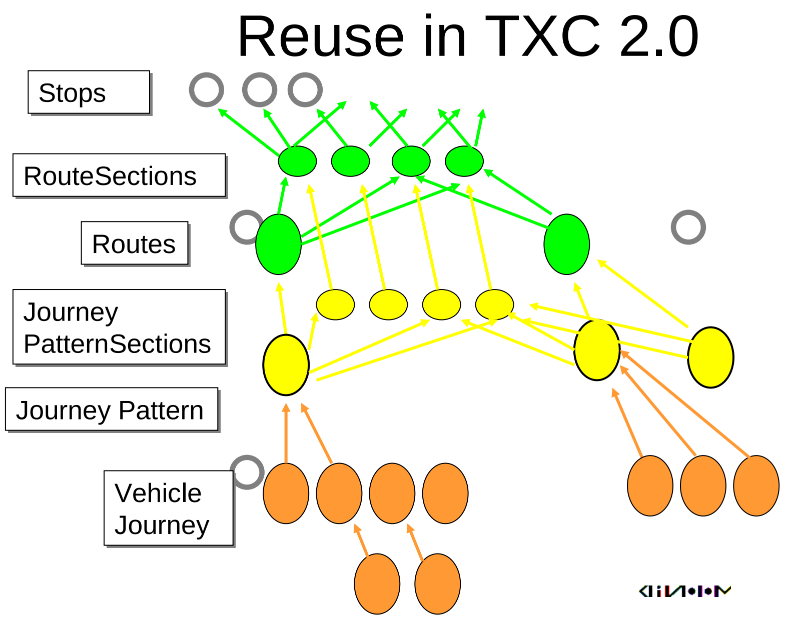


Figure 3-60 – Reuse of Elements

3.16.3.2 Inefficiencies in TransXChange

Although inheritance, default values and reuse can be used to optimise document content, *TransXChange* is not a fully optimised representation, and has a number of data redundancies in its representation. In particular:

- Start and end stop usages are repeated on every successive link in a journey pattern and vehicle journey, including the stop point reference on the journey pattern usage.
- Start and end stop points are repeated on both route links and journey pattern links.

3.16.3.3 Use of Sections

Route sections allow implementers to reuse a sequence of stops and links in more than one route. Journey pattern sections allow the corresponding sequence of timing links to be reused. *Figure 3-61* shows an example of a service comprising three named lines (*Line 54 Line 54A* and *Line 12*). The lines are made up of four routes, containing five sections that group the eleven different links of the network. Two of the sections (*S1 & S2*) are reused in two different routes.

- *Line 54*
  → R1 = S1(L1, L2) + S2(L3 + L4 + L5 + L6)
  → R2 = S1(L1, L2) + S4(L3 + N1 + N2 + L6)
- *Line 54A*
  → R3 = S3(M1) + S2(L3+L4+L5+L6)
- *Line12*
  → R4 = S3(M1) + S5(R1, R2)

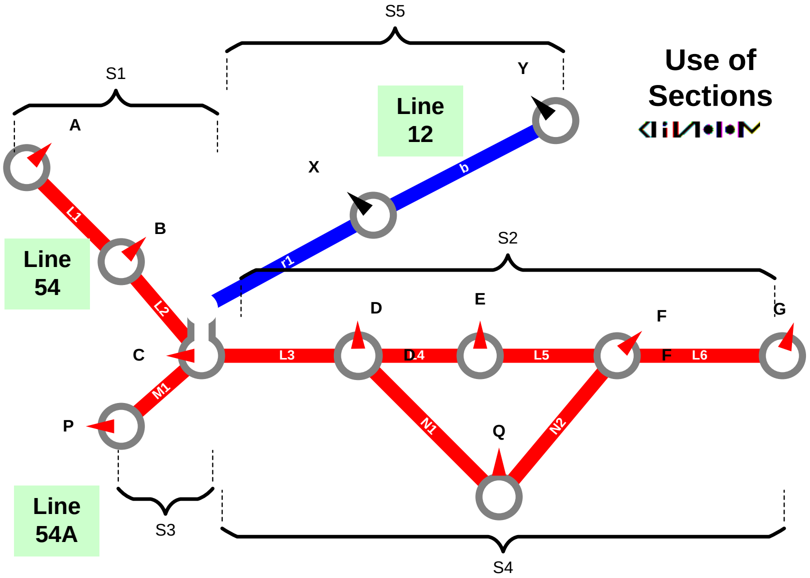


Figure 3-61 – Example of Sections

3.16.4 Presenting Schedules in Timetables

TransXChange is primarily concerned with the representation of schedule data for exchange between different computer systems, and is not intended to address all the additional requirements for presenting schedules as published representations for the public. However it is possible to transform a *TransXChange* schedule into a matrix timetable format automatically, adopting a specific order for showing the stops. Rendering the journeys in a tabular format is valuable because it allows a *TransXChange* document to be validated by human inspection against the originating and published formats.

The *TransXChange Publisher* provides an example of a matrix rendering, which follows a conventional mapping:

- ***VehicleJourney*** instances generally correspond to columns.
  - Each ***VehicleJourney*** instance can have a ***VehicleJourney*** / ***Note*** associated with it.
  - Each ***VehicleJourney*** can have an ***OperatingProfile*** to specify operational time information specific to it in a quantitative structure.
  - If a ***Frequency*** is specified, one or more additional columns may be interpolated to indicate the repeating journeys.
  - If a ***Frequency*** and the same ***EndTime*** is specified for more than one journey, one or more journey columns may be merged to create a single frequency group. See 3.18.8.4.
  - ***VehicleJourney*** are ordered as columns across the matrix in the same order as they are declared in the document. Normally they should be sorted into time order.
- ***JourneyPatternTimingLink / VehicleJourneyStopUsage*** instances generally correspond to each individual row.
- ***VehicleJourneyTimingLink / VehicleJourneyStopUsage*** instances generally correspond to cells for each individual row.
  - Each ***From*** / ***VehicleJourneyStopUsage*** corresponds to a departure. Normally the departure is shown for all stops of the route except the last.
  - Each ***To*** / ***VehicleJourneyStopUsage*** corresponds to an arrival. Normally arrivals are only shown for the last stop.
  - If the arrival and departure time is different at a stop, two separate rows for arrival and departure will be shown.
  - The stop rows will be ordered down the page in the same order that ***VehicleJourneyStopUsage*** instances appear in the Journey pattern of the Vehicle Journey (unless overridden by a ***SequenceNumber)***.
    - To collate different journeys that follow different journey patterns into a single matrix, the publisher compiles a list of all the stops of all the journey patterns, in order of use. The stops of each vehicle journey are aligned against this list, leaving an empty cell for any stop that is not visited by a particular journey. Thus for example, if two journey patterns A-B-F and A-C-D-F are combined as list, these will be collated as A-B-C-D-F, resulting in column entries A-B-()-()-F for the first and A-()-C-D-F for the second.
    - Within this overall ordering, where there are stops that are specific to particular journey patterns these will be ordered according to the passing time. Thus for example, consider two vehicle journeys using separate patterns A-B-D and A-C-D, which have passing times at stop A(t_1)-B(t_4)-D(t_6) and A(t_2)-C(t_3)-D(t_5) where t_n indicates the relative time. If these are combined as list, these will be ordered A-C-B-D, rather than say A-B-C-D because stop C is visited earlier (t_3) than stop B (t_4).
    - As a further refinement to the overall ordering, the publisher uses the grouping of stops given by the ***JourneyPatternSection*** to sequentially order a series of stops in succession that are a route variants used only by certain journeys, rather than the strict relative time. Thus for example, if two journey patterns A-[B-C-D]-F and A-[P-Q-R]-F are combined as list these will be ordered as A-B-C-D-P-Q-R-F, rather than say as A-B-P-C-Q-D-R-F, regardless of the relative passing times at BCD and PQR. This results in more readable columns entries of A-B-C-

D-()-()-()-F and A-()-()-()-P-Q-R-F, rather than say A-B-()-C-()-D-()-F and A-()-P-()-Q-()-R-F.

- A ***SequenceNumber*** attribute can be specified on individual ***JourneyPatternStopUsage*** instances to suggest a preferred sort order of stops for presentation. When listing the stops as rows in a matrix, the explicit number overrides the default traversal sequence that will be otherwise assumed for publication. Note that each vehicle journey is still traversed by a bus in the actual order of its links regardless of any ***SequenceNumber*** instances.

3.16.4.1 Using a Sequence Number

The ***SequenceNumber*** attribute on individual ***JourneyPatternStopUsage*** instances allows you to control the ordering of stops in tabular presentations.

1. Every stop usage of a journey pattern timing link can be allocated a sequence number (i.e. both the departure-from-stop end, and arrival-at-stop end).
2. Either all of the stop usages (*'explicit numbering'*), or none of the stop usages (*'implicit numbering'*) of a complete journey pattern should be numbered. (N.B. If some are numbered and some are not, indeterminate effects may occur in applications that make use of the ***SequenceNumber***.)
    - For *implicit* stop numbering, each stop usage is consequentially numbered in order of traversal of the timing links of the route and journey pattern. (Note that journey pattern timing links must in any case visit the same stops in the same order as the route links of any route that the journey pattern references).
    - For *explicit* stop numbering, each stop usage is consequentially numbered in the implementer’s preferred presentation order. The actual traversal of the journey timing links for the computation of passing times still follows the sequence of the links of the journey pattern, even if the stops are sequenced in a different order by means of a ***SequenceNumber***.

For example, the *TransXChange Publisher* uses the ***SequenceNumber*** as follows:

1. The publisher builds a matrix by creating a line for each vehicle journey stop usage (i.e. arrival and departure, of each vehicle journey in the service), and sorting them all into stop sequence order.
   - If there are several different underlying journey patterns (i.e. routes) making up the overall service, giving rise to overlapping (or even completely disjoint) sets of stops, the publisher takes the combined set of all stop usages; if the same stop usage appears with the same sequence number in multiple journey patterns, then it is shown as the same stop row.
2. For each column, i.e. vehicle journey, the stop passing times for each stop are computed in the order of traversal of the timing links; times are only shown in cells for the stops that are visited.
3. If the arrival and departure usages for the same stop appear on consecutively lines on the matrix, they can be shown as a single line, showing just the departure time.

3.16.4.2 Example of a Timetable using StopSequence

Figure 3-62 shows a service with two alternate routes (*R1 & R2*) over six stops (labelled alphabetically '*A'* to '*F*') and which are labelled; line '*1C'*, which runs '*A-B-C-E-F'*, and line '*1D'*, which runs '*A-B-D-E-F'.*

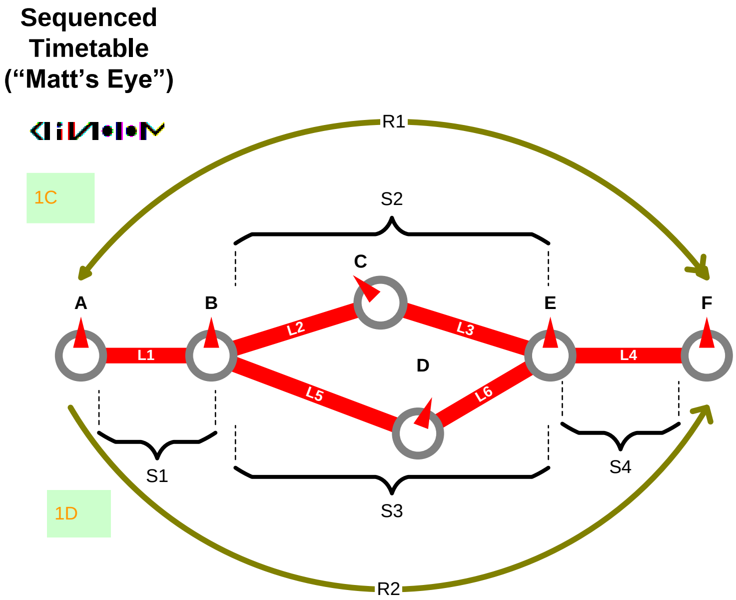


Figure 3-62 – Example: Use of Stop Sequencing

In the published timetable for the service, the preferred presentation might be to show the two journeys aligned on all similar stops – see *Table 3-11.*

	Journey		*SequenceNumber*
	1C	*1D*	*#*
A	10:00	11:00	*1*
B	10:05	11:05	*2*
C	10:10	-------	*3*
D	-------	11:12	*4*
E	10:15	11:17	*5*
F	10:20	11:22	*6*

Table 3-11 – Example: Eye Timetable with Explicit Stop Sequencing

To specify the above presentation we might do the following:

- Break the two routes down into four sections containing route links as follows:
  - R1 = RS1(RL1) + RS2(RL2, RL3) + RS4(RL4)
  - R2 = RS1(RL1) + RS3(RL5, RL6) + RS4(RL4)
- Define a journey pattern, JP1, over route R1, specifying a preferred stop sequence *n* for each end of each timing link:
  - • JP1 = → R1

    [JS1(→RL1: JPTL1[*5mn*, from:1, to: 2])
    + JS2(→RL2: JPTL2[*5mn*, from:2, to: 3]. →RL3: JPTL3[*5mn*, from:3, to: 5])
    + JS4(→RL5: JPTL4[*5mn*, from:5, to: 6])]
- Define a journey pattern, JP2, over route R2: also specifying a preferred stop sequence:
  - JP2 = → R2

    [JS1(→RL1: JPTL1[*5mn*, from:1, to: 2])
    + JS3(→RL5: JPTL5[*7mn*, from:2, to: 4], →RL6: JPTL6[*5mn*, from:4, to: 5])
    + JS4(→RL4: JPTL4[*5mn*, from:5, to: 6])]

As a comparison, Figure 3-67 shows the default ordering that would be used by the *TransXChange Publisher* if no sequence guidance was given. Note this is a difference of presentation, not representation – in the underlying *TransXChange* document, each individual vehicle journey is still correctly ordered as to its sequence of visiting the stops by virtue of its journey pattern. If wait times had been specified, then arrival and departure would be distinct.

Stop	1C	1D
A	*Dep 10:00*	*Dep 11:00*
B	*Dep 10:05*	*Dep 11:05*
C	*Dep 10:10*	-
E	*Arr 10:15*	-
B	-	*Dep 11:05*
D	-	*Dep 11:12*
E	-	*Arr 11:17*
E	*Dep 10:15*	*Dep 11:17*
F	*Arr 10:20*	*Arr 11:20*

Table 3-12 – Example: Eye Timetable with Implicit Stop Sequencing

3.16.5 Grouping Journeys within a Timetable

TransXChange includes mechanisms for specifying how journeys should be grouped within a service. This grouping can be used by the publisher and other tools to present journeys grouped by day type or other criteria. A text description can be associated with each of the groupings. In some circumstances, for example for large but sparsely populated or complex timetables moving one or more exceptional journeys to a separate matrix can greatly reduce the overall size of the matrix.

By Default six ‘built-in’ Journey Groupings are assumed for each Service:

- **Outbound**, **Monday to Friday**, **Saturday** and **Sunday.**
- **Inbound**: **Monday to Friday**, **Saturday** and **Sunday.**

These built-in journey groupings do not have to be explicitly specified unless you wish to associate a description or note with them.

Additional ***CustomJourneyGroupings*** may be added (+TXC v2.4). Each ***CustomJourneyGrouping*** must explicitly specify the ***VehicleJourney*** instance to be included. Any custom journeys will be omitted from the built-in Journeys.

3.16.5.1 Journey Grouping Model Overview

Figure 3-67 gives an overview of the ***JourneyGrouping*** elements.

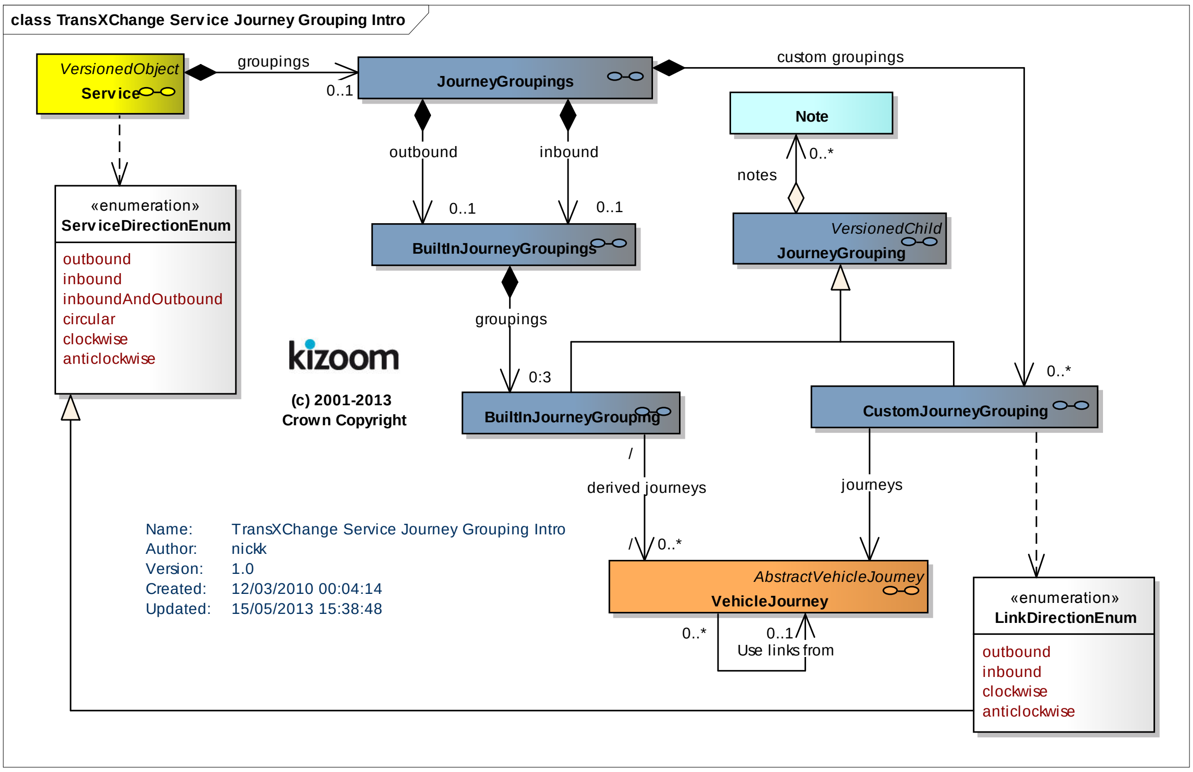


Figure 3-63 – UML Diagram of Service Journey Grouping elements - Introduction

3.16.5.2 JourneyGrouping Elements

Figure 3-64 shows the ***JourneyGrouping*** elements in more detail.

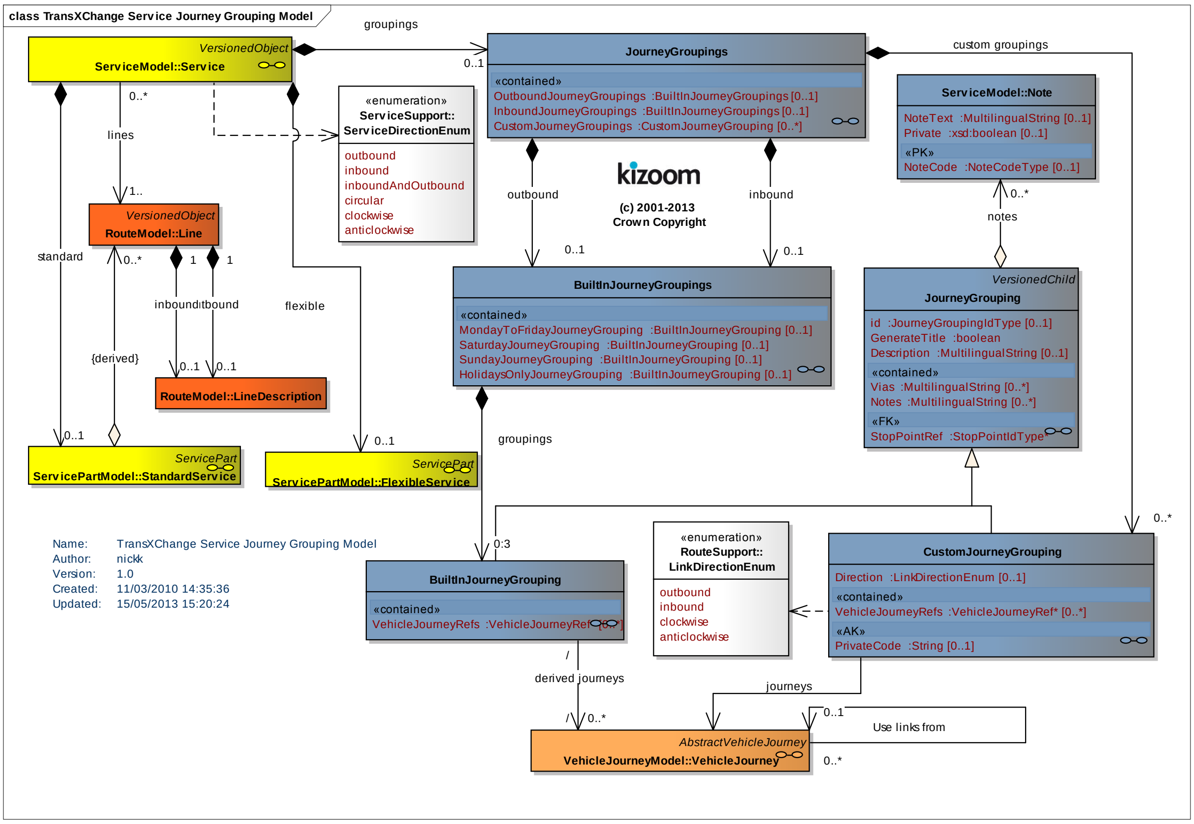


Figure 3-64 – UML Diagram of Service Journey Grouping elements – Details

3.16.5.3 JourneyGrouping Model Hierarchy

Figure 3-65 shows the inheritance hierarchy for the ***JourneyGrouping*** model elements.

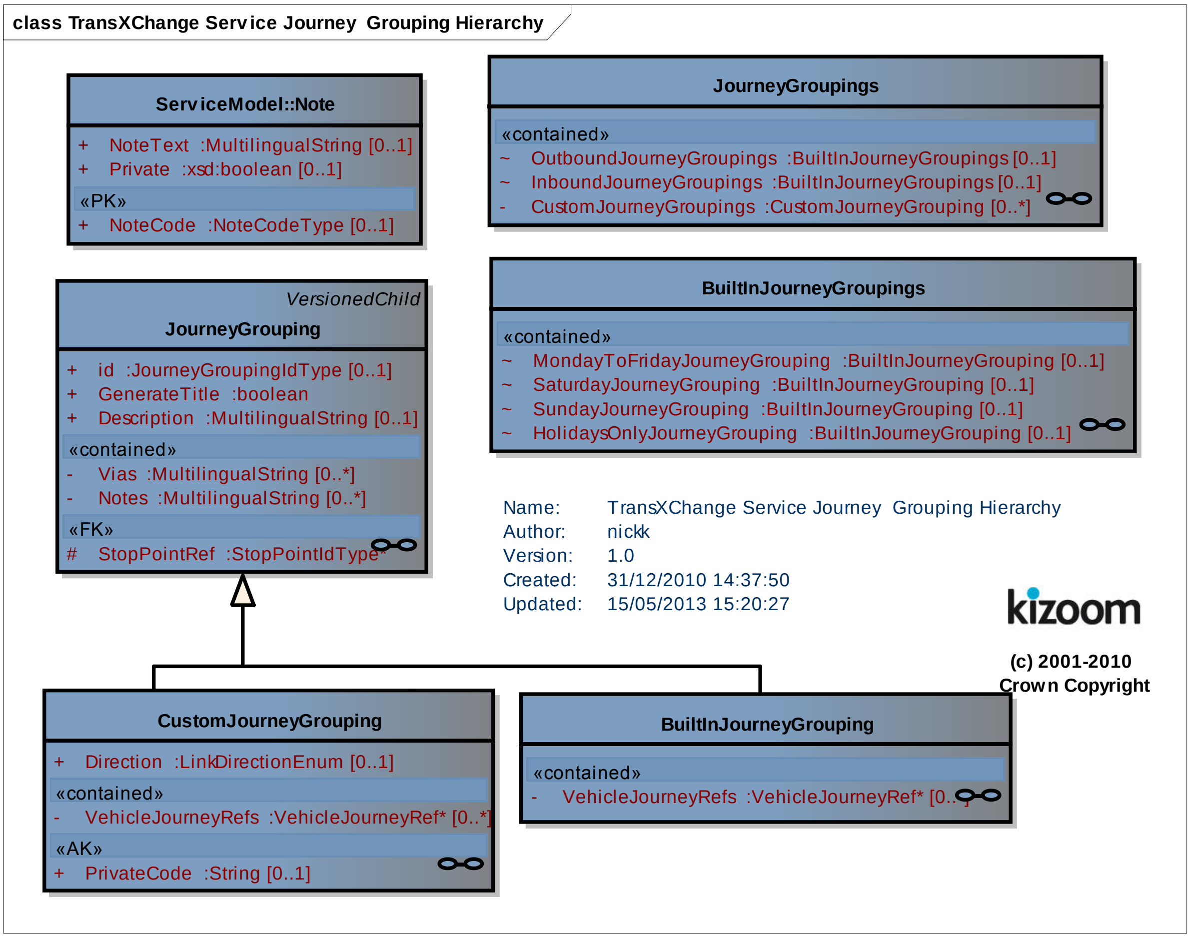


Figure 3-65 – UML Diagram of JourneyGrouping Element Hierarchy

3.16.6 Specifying services that run before or after midnight

The operating profile of a journey states the day or days of the week on which it runs, for example, '*Monday to Friday', 'Monday to Saturday', Sunday*, etc. This is normally *the day in which the departure time of the journey falls*

Sometimes however OPERATING DAY may be different, for example services for a given operating day might run from 2 am to at 2 am). For example:

1. **Late night service.** A service that starts in the early morning after midnight but that is part of the previous evening's service (i.e. operating day). For example, a service that runs at *01:10 Tuesday to Saturday* nights as part of a Monday to Friday service
2. **Early morning service.** A service that starts late at night before midnight but that is part of the next day's service. For example, a service that runs at *23:55 Sunday to Thursday* nights as part of a Monday to Friday service.

This can be achieved by means of the ***DayShift***. The day shift the times of the journey forward or back by twenty four hours. :

- A negative ***DayShift*** indicates a journey time is in the previous day and should be shown with a footnote '** Previous day'.
- A positive ***DayShift*** causes indicates a journey time is in the next day and should be shown with a footnote '** Next day'.

Journeys should be placed in a TransXChange document in their true temporal order, that is (i) negatively shifted journeys in departure time order, (ii) unshifted journeys in departure time order and (iii) positively shifted journeys in departure time order.

3.17 DayShift Examples

3.17.1 Late night Weekday service – Positive DayShift

Table 3-13 shows an example of a *Monday to Friday* service where there is a post-midnight early morning journey (#3) that actually takes place on *Tuesday Saturdays at 00:30*.

3.17.1.1 Actual Journeys

top	**#1**	**#2**	**#3**
DayOfWeek	**MF**	**MF**	**MF**
Actual days	MTWTF--	MTWTF--	-TWTFS-
			DayShift +1
A	20:30	21:30	00:30
B	20:40	21:40	00:40
C	21:10	22:10	01:10

Table 3-13 – Positive DayShift example

3.17.1.2 As Shown in Beds with Positive DayShift

Journey *#3* is given a positive ***DayShift*** of '*+1'*, which has the following effect:

- Indicates to journey planners etc that he service is actually in the next day
- Causes it be flagged *with a* foot note indicating it runs on the next day.

Stop	**#1**	**#2**	**#3**
A	20:30	21:30	*00:30*** Next Day*
B	20:40	21:40	*00:40*** Next Day*
C	21:10	22:10	*01:10*** Next Day*
Note			*Tuesday to Saturday*

Table 3-14 – Positive DayShift example - effect

3.17.2 Early morning Weekday Service – Negative DayShift

Table 3-15 shows an example of a *Monday to Friday* service where there is a pre midnight late evening journey (#1) that actually takes place on *Sunday to Thursdays* at *23:50*.

3.17.2.1 Actual Journeys

Stop	**#1**	**#2**	**#3**
DayOfWeek	**MF**	**MF**	**MF**
Actual days	MTWT--S	MTWTF--	MTWTF--
	DayShift - 1		
A	23:30	10:30	21:30
B	23:40	10:40	21:40
C	00:10	11:10	22:10

Table 3-15 – Negative DayShift example:

3.17.2.2 As shown in Beds with Positive DayShift

Journey #3 is given a negative ***DayShift*** of *'-1'*, which has the following effect:

- Indicates to journey planners etc that he service is actually in the previous day
- Causes it be flagged *with a* foot note indicating it runs on the previous day.

Stop	**#3**	**#1**	**#2**
A	*23:30*Previous day*	20:30	21:30
B	*23:40*Previous day*	20:40	21:40
C	*00:10*	20:50	21:50
Note	*Sunday to Friday*		MF

Table 3-16 – Positive DayShift example: Monday to Friday Bed

3.17.3 Associating Operational Data with a Timetable

TransXChange provides several means of associating different types of operational data with the elements of a timetable (See Figure 3-66). For example,

- ***JourneyPatterns*** & ***VehicleJourneys*** may be associated with an ***Operational*** element that specifies a ***Block***, ***VehicleType*** or ***TicketMachine*** for a journey.
- ***The VehicleType*** may include basic accessibility data such as whether the vehicle is considered accessible, and some further details of the ***VehicleEquipment***.
- A ***TimingLink*** may specify a ***DutyCrew*** for a link.

3.17.3.1 Operational Data Model Introduction

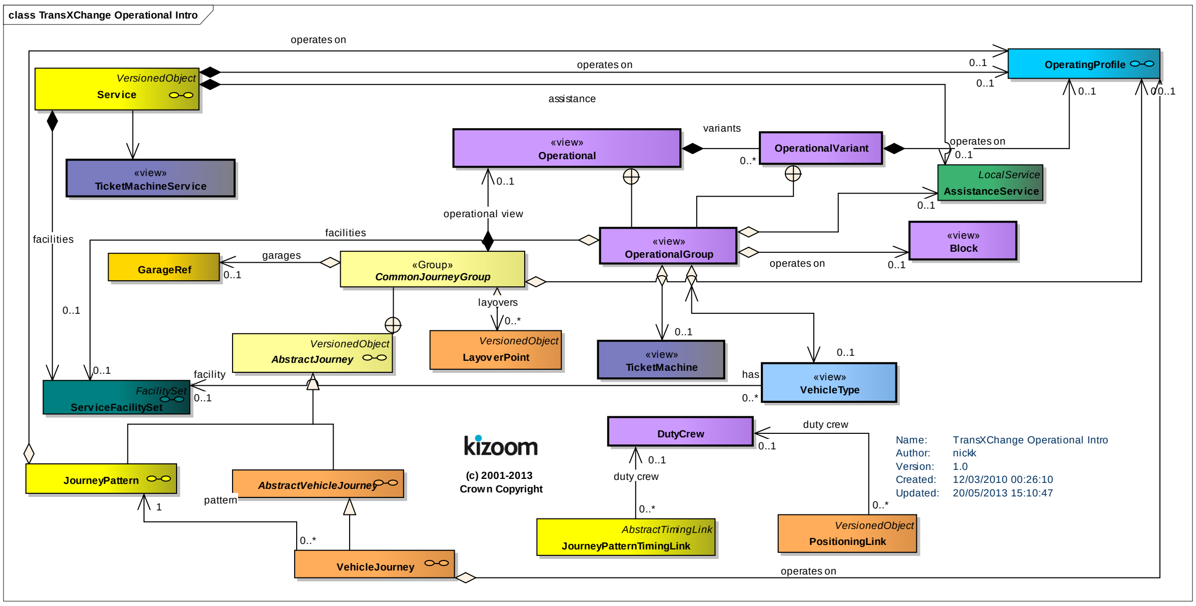


Figure 3-66 – UML Diagram of Operational data elements: Introduction

3.17.3.2 Operational Data Model Details

Figure 3-67 shows the details of the Operational model.

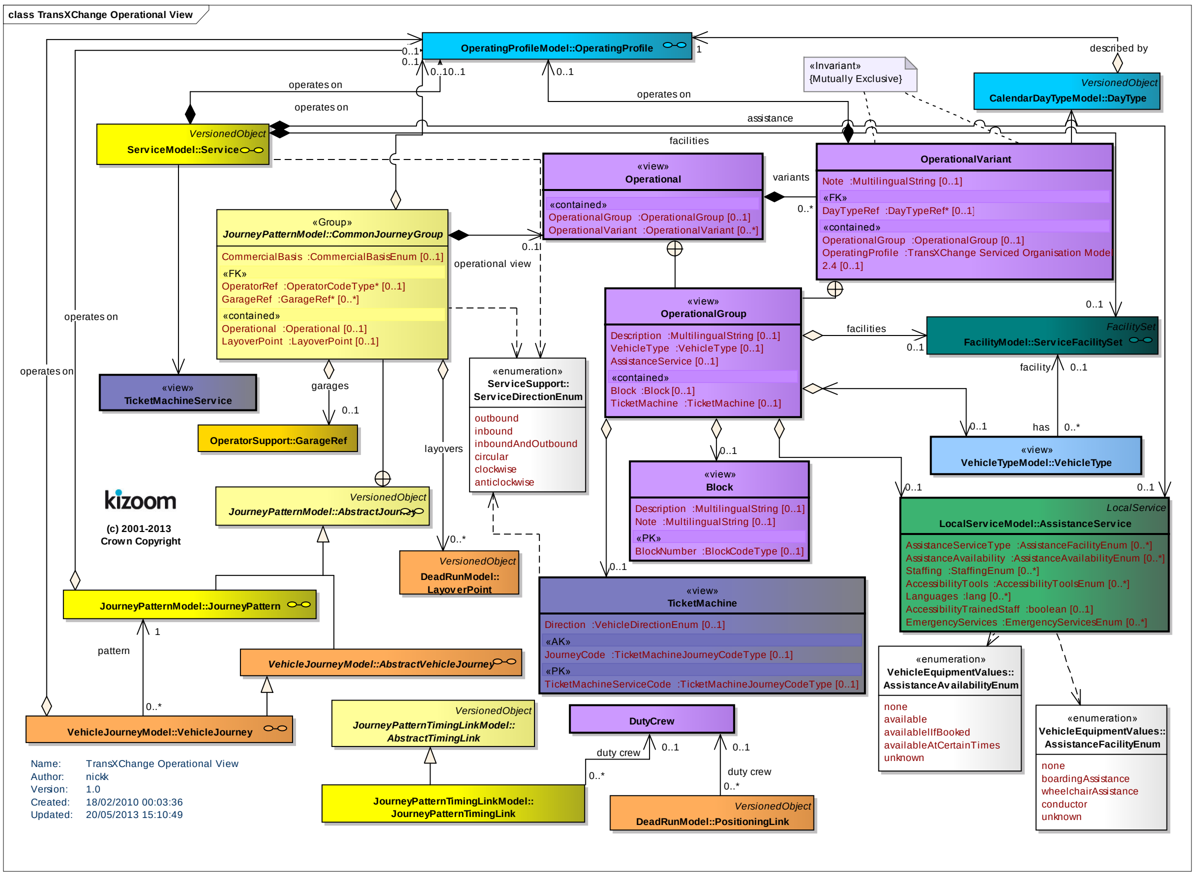


Figure 3-67 – UML Diagram of Operational data elements: Details

3.17.4 Operational Model Hierarchy

Figure 3-68 shows the inheritance hierarchy for the Operational model elements.

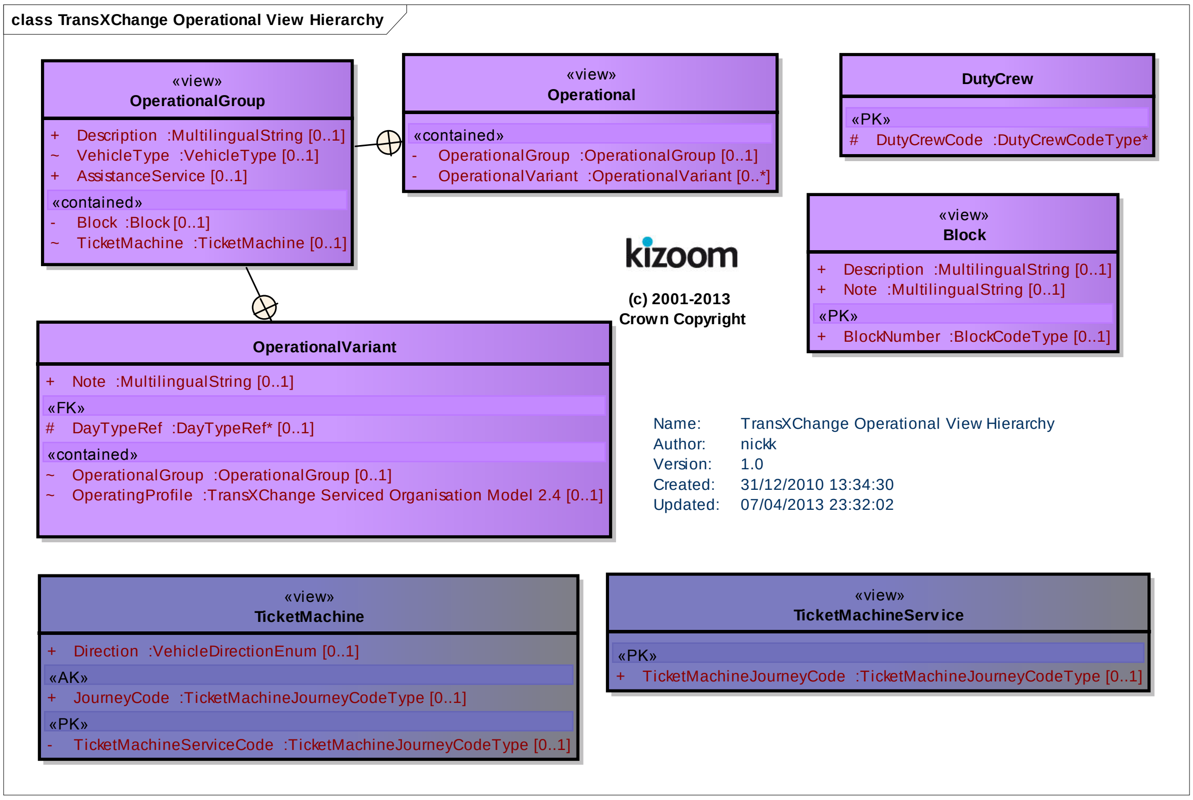


Figure 3-68 – UML Diagram of Operational Element Hierarchy

3.17.4.1 Operational Data Model Support Types

Figure 3-69 shows the additional data types used by the ***Operational*** model elements.

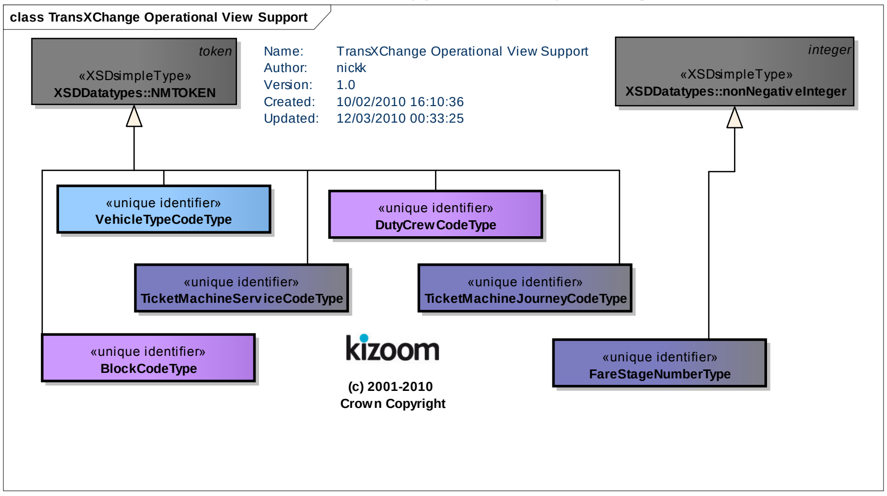


Figure 3-69 – UML Diagram of Operational Support types

3.17.5 Vehicle Equipment Model (+TXC v2.4)

Figure 3-70 shows the vehicle equipment data types that may be used to specify the accessibility facilities of a vehicle. These are based on the IFOPT/NeTEx model.

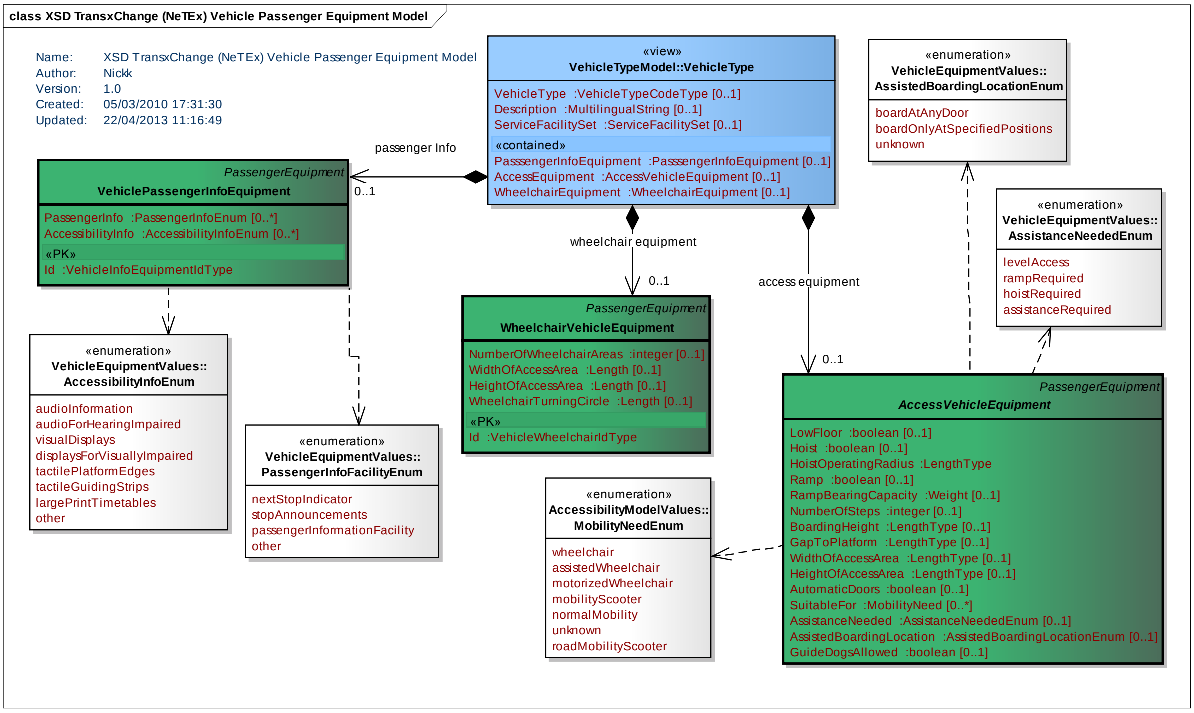


Figure 3-70 – UML Diagram of VehicleEquipment Types

3.17.6 Assistance Service Model (+TXC v2.5)

Figure 3-71 shows the ***AssistanceService***, a type of local service "Equipment" that may be used to specify details and conditions about assistance available for a ***VehicleJourney***. These are based on the NeTEx model.

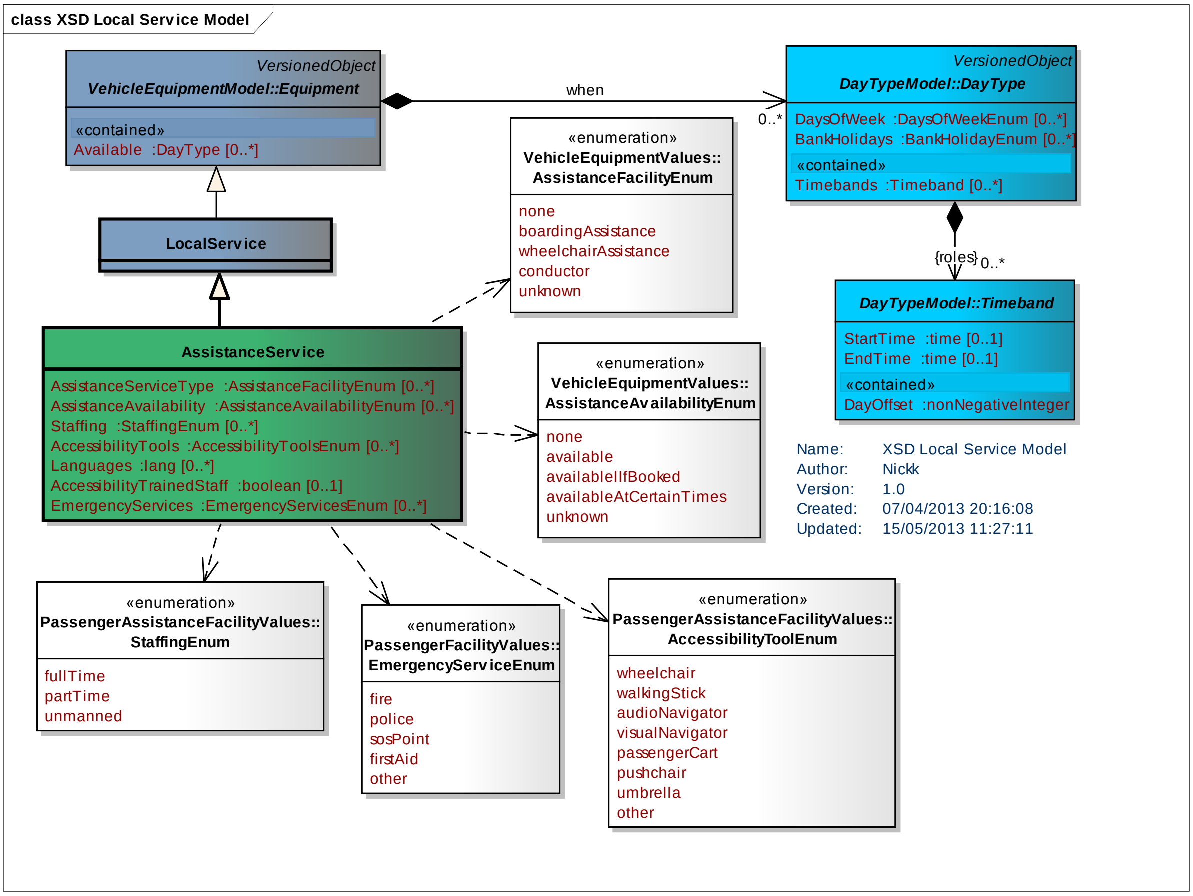


Figure 3-71 – UML Diagram of Assistance Service Types

3.17.7 ServiceFacilitySet Model (+TXC v2.5)

*A **ServiceFacilitySet***, Figure 3-71, can be used to list facilities available on a service or specific **JourneyPattern** or ***VehicleJourney***. The facilities are just list of named properties – any detailed attributes of a specific facility are described by a corresponding ***Equipment*** element. The values supported are a subset of those found in the NeTEx model.

Each facility value specified at the Service level apply to all journeys unless overridden on a specific **JourneyPattern** or ***VehicleJourney***.

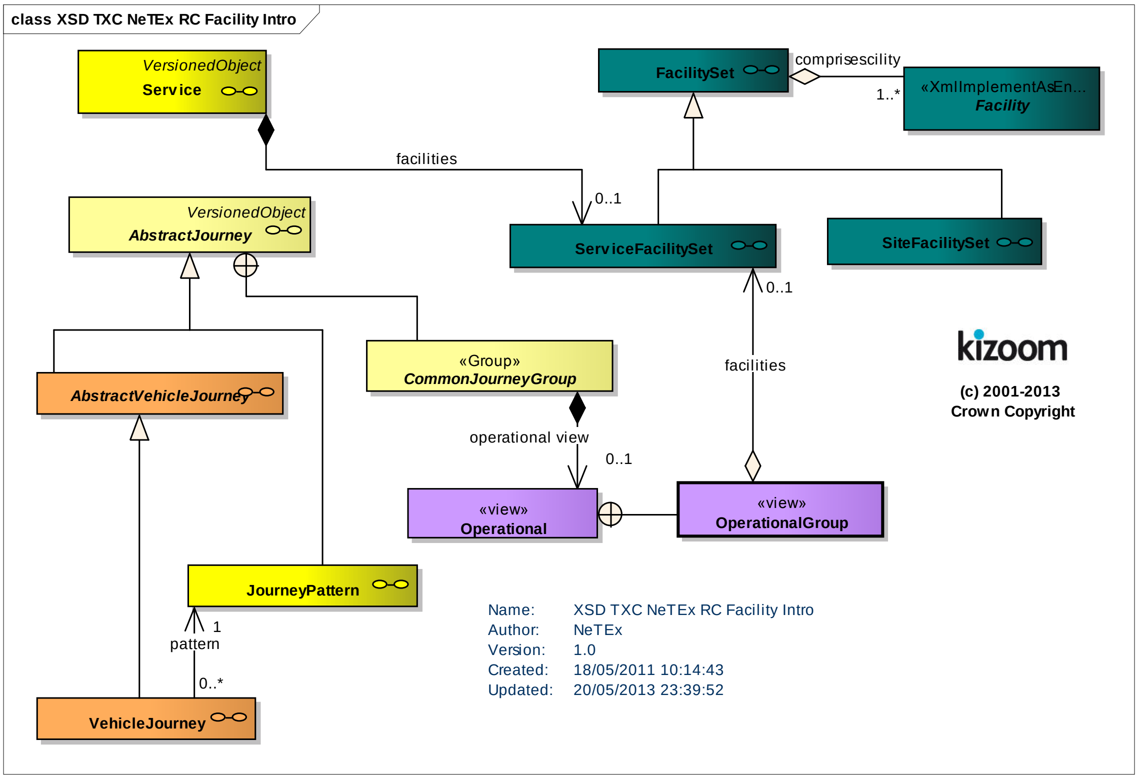


Figure 3-72 – UML Diagram of ServiceFacilitySet Introduction

3.17.7.1 ServiceFacilitySet Data Model Details

Figure 3-73 shows the details of the ***ServiceFacilitySet*** model.

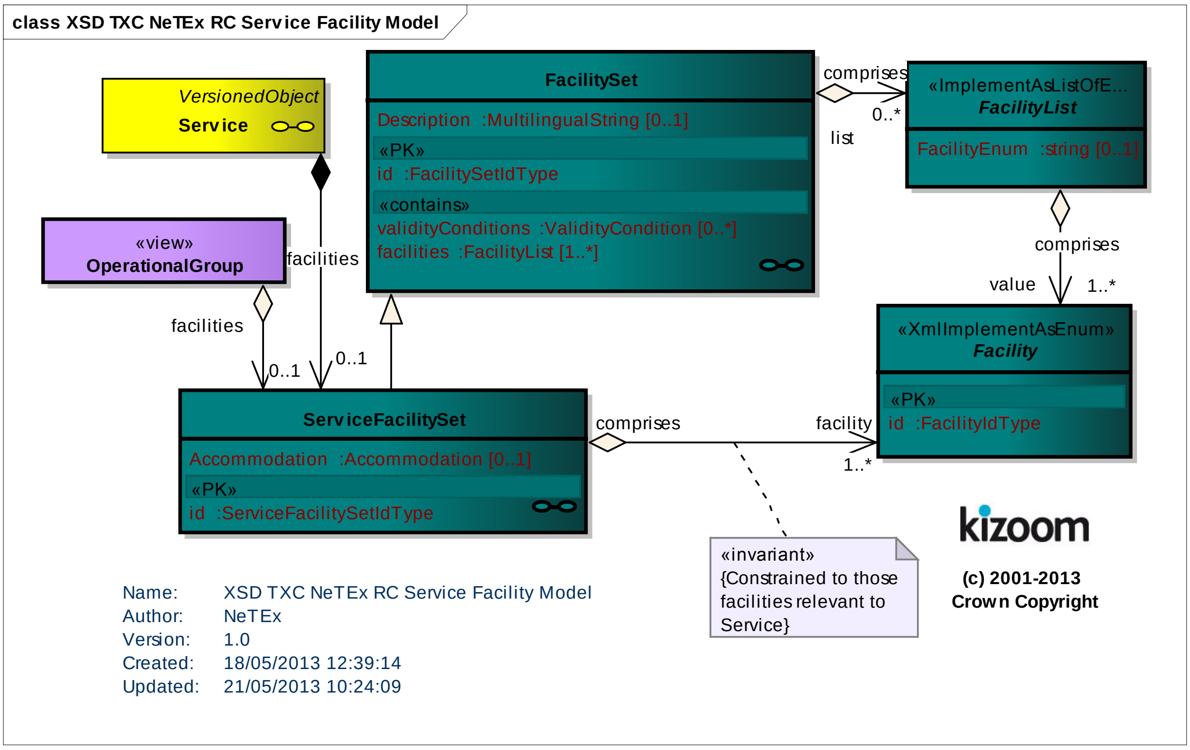


Figure 3-73 – UML Diagram of ServiceFacilitySet data elements: Details

3.17.7.2 ServiceFacilitySet Data Model – Common values

Figure 3-74 shows some the ***Facility*** values that may be specified for a ***ServiceFacilitySet***. These include ***FareClass, CateringFacility, NuisanceFacility, PassengerCommsFacility, PassengerInformationFacility, SafetyFacility,*** and ***SanitaryFacility***

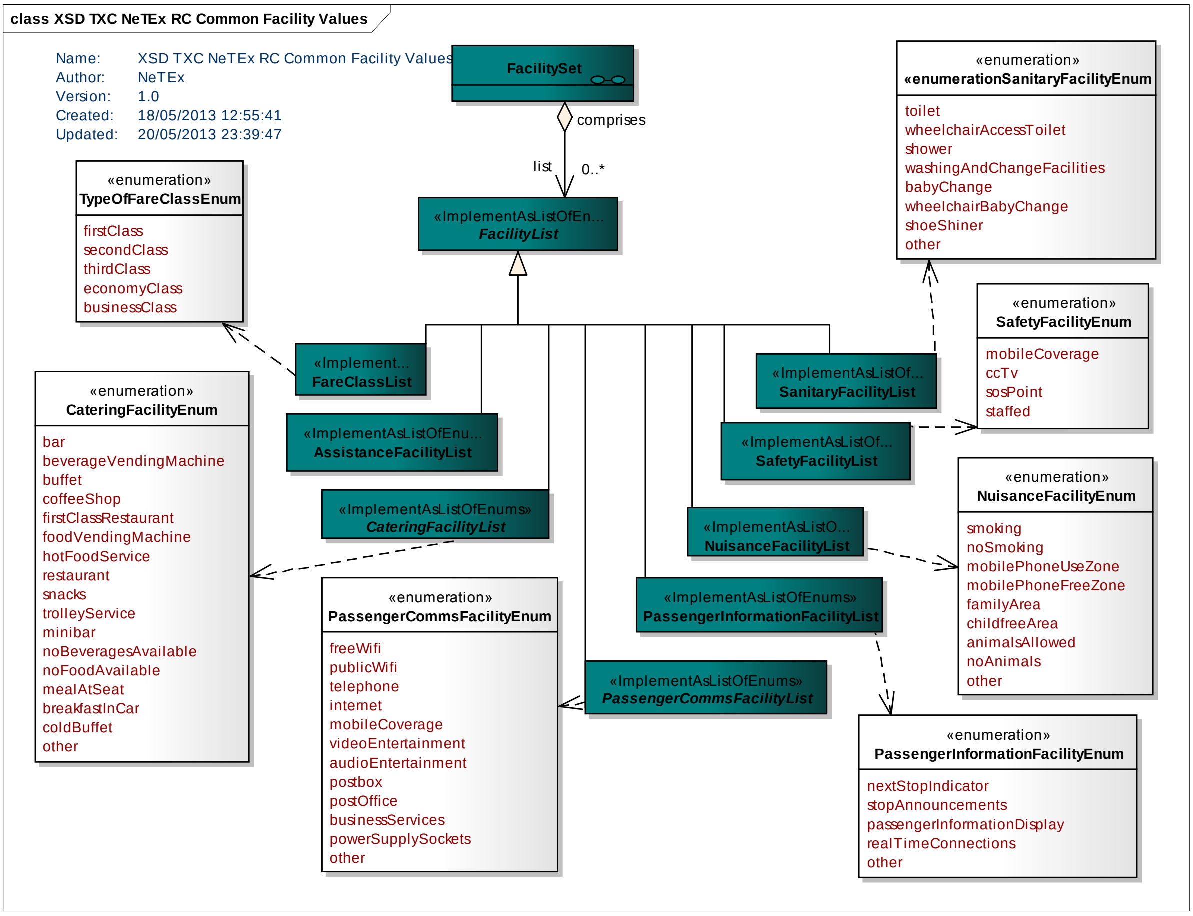


Figure 3-74 – UML Diagram of ServiceFacilitySet: Common values

3.17.7.3 ServiceFacilitySet Data Model – Specific values

Figure 3-75 shows three further lists of values that may be specified for a ***ServiceFacilitySet***: ***AccommodationFacility***, ***LuggageCarriageFacility*** and ***ServiceReservationFacility***.

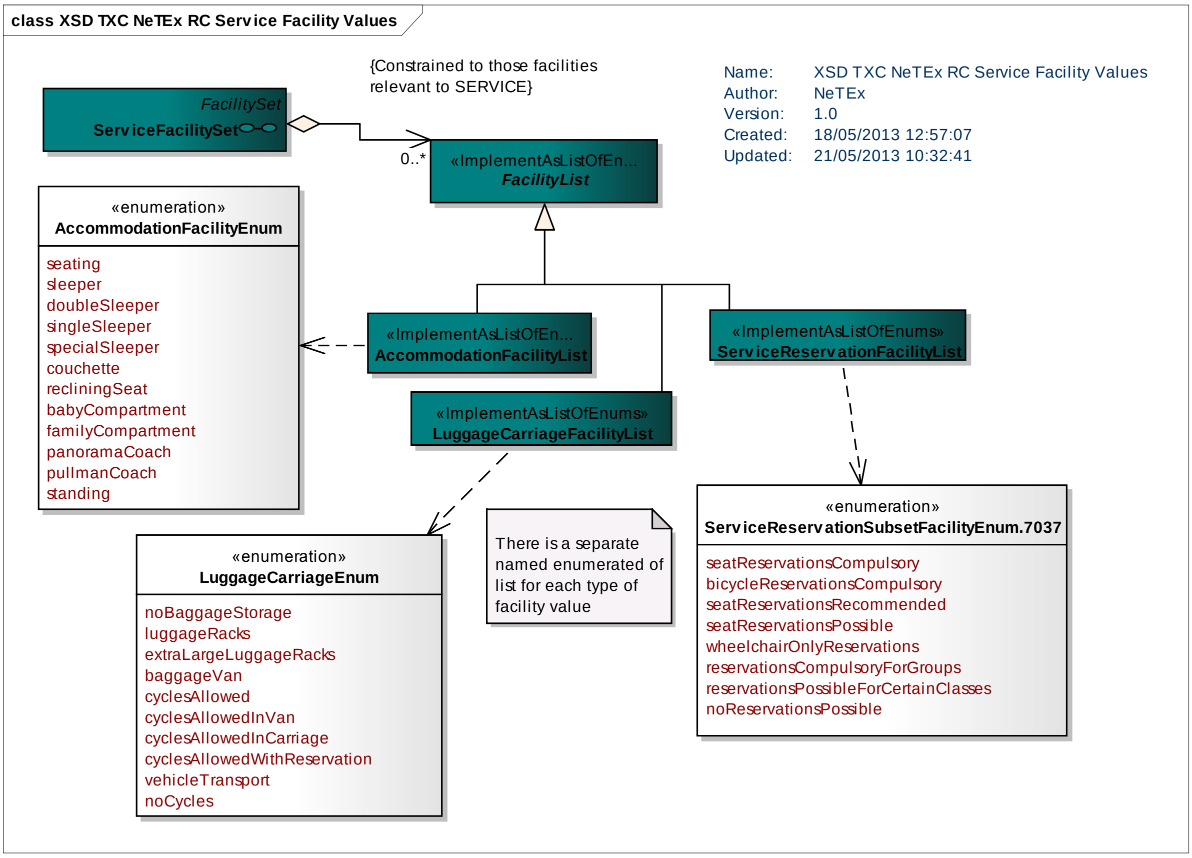


Figure 3-75 – UML Diagram of ServiceFacilitySet: Specific values

3.17.8 Service Descriptions

The TXC schema allows descriptive elements to be declared that describe where a service runs:

(a) Each **Service** may have a **Description**. This is non-directional. In a registration there is only a single Service. In a general document there may be several Services.

- For each **Service** an **Origin** place name, **Destination** place name and a list of **Via** place names can also be specified, these can be used to generate text descriptions for the Service.
- The ***Service / Description*** is published in the Service particulars.

(b) Each **Line** of a **Service** may have an ***OutboundDescription*** and an ***InboundDescription*** (added in TXC v2.4).

- Each description can have an **Origin** place name, **Destination** place name and a list of **Vias** place names can also be specified. These can be used to generate directional text descriptions for the Service. There might be more than one line for a service, and so multiple line descriptions.

(c) Each **Route** may have a Description. There may be many Routes, each with one or more journey patterns and hence one or more vehicle journeys associated with it. It is published in a list of route names as part of the service particulars. Separate Routes can (and often are) be used each direction, so it can be used to get a separate description for each direction. However the Route Description cannot be related to specific journeys or matrix beds, since a matrix bed may reference more than one route.

- The ***Route / Description*** is published in the Service particulars.

(d) Each **JourneyPattern** may have a **Description**. This is directional. There may be many journey patterns, each with one or more vehicle journeys in a service. Since there may be more than one journey pattern associated with a single matrix bed, one cannot necessarily determine which journey pattern (and hence description) to use for the bed.

- The value is not currently published.

(e) Each **VehicleJourney** may have a **Description**. This is directional – It applies to the direction of the individual journey. There will be many vehicle journeys associated with a given matrix bed.

- The value is not currently published.

(f) A **Description** element on each **JourneyGrouping**. (+TXC v2.4) There will be many vehicle journeys associated with a given Journey Grouping.

- The value is published as the title for each matrix bed. (TXC 2.4)

3.18 Modelling Operational Days

TransXChange has rich (and complex!) capabilities for specifying the operational days and times of a bus service for both regular running, and for exceptional days. We introduce these capabilities here. For further details, see the descriptions of individual schema elements. For an overall summary of how to combine date conditions, see also *Section 13* ‘Integrity Rules’.

3.18.1 Specifying When the Service Operates – Summary

The ***OperatingProfile*** specifies when a bus service runs as a set of day properties, including both the types of days (e.g. *Monday* to *Friday*) on which the service normally runs; and what happens on special days such as Bank Holidays, and can also describe any exceptional periods of operation.

- An overall ***OperatingPeriod*** can also be specified at the service level. This can be open ended.
- Default ***OperatingProfile*** values can be specified at the ***Service*** level, and be overridden at both the ***JourneyPattern*** level and on individual ***VehicleJourney*** instances.

- Validity periods can also be specified for the operation of ***JourneyPatternInterchange*** instances, constraining the availability of interchanges between specified ***VehicleJourney*** instances.

- Operational times may also be specified in terms of the working days of a ***ServicedOrganisation*** – see later below

An operational ***Calendar*** can be used to relate the Day Types to specific dates in a given year.

Figure 3-76, in UML class diagram notation, gives a high-level view of the main elements and relationships concerned with operational days.

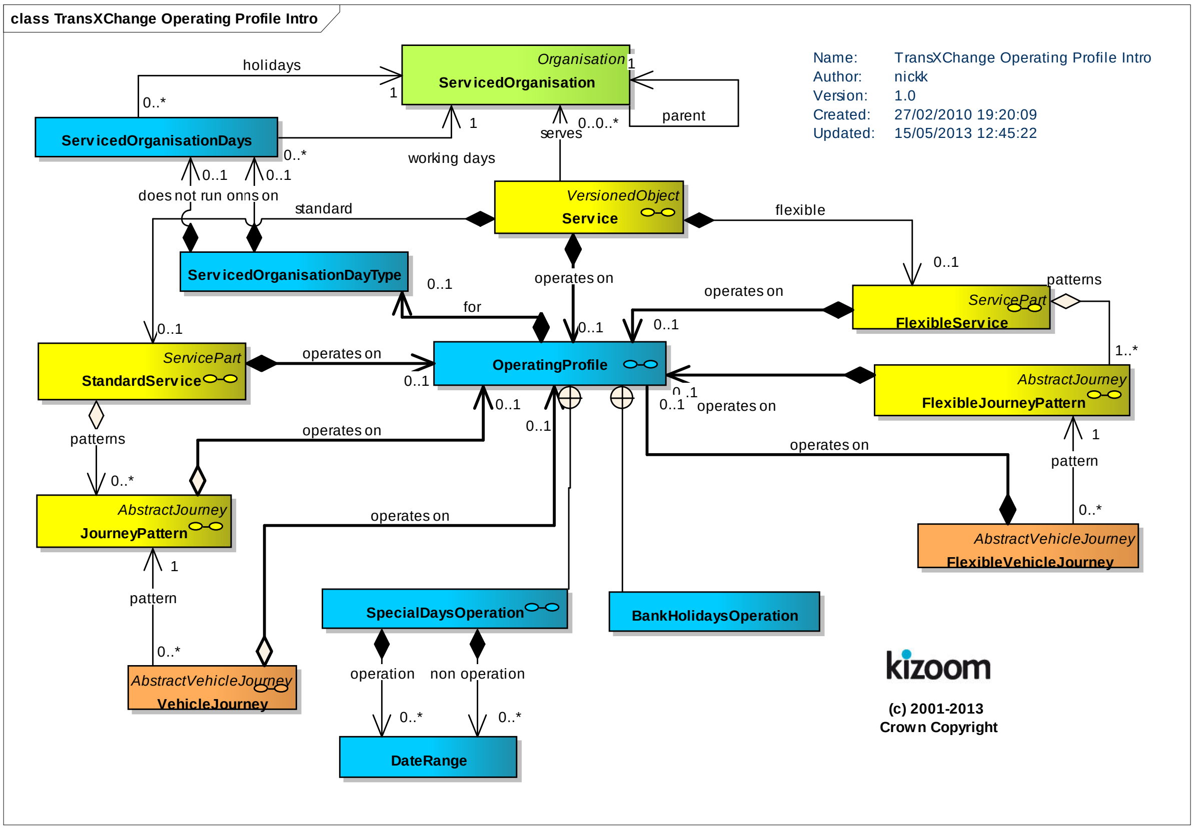


Figure 3-76 – UML Diagram Overview of Operational Times

3.18.2 Regular Operation – OperatingProfile

The ***OperatingProfile / Normal*** group specifies the normal operating day types of a service. It can be made up of three elements, as shown in the UML structure diagram in *Figure 3-77*:

- The types of day (***RegularDayType***) on which the service runs; for example, '*Monday to Friday', 'Sunday',* or *'Wednesday and Saturday.'*
- The weeks of the month on which the service is operated for the given day types; for example, 'first and third weeks of the month'. The ***PeriodicDayType*** further qualifies the ***RegularDayType***, (for example, a market service that might run '*Wednesdays and Saturdays, first and third weeks of the month'*).
- The holiday or working day types of the serviced organisation for which the service runs (for example, '*term times for City of London School for Girls'*) – see ***ServicedOrganisation*** in Section 3.18.4 below. The ***ServicedOrganisationDayType*** further qualifies the periodic and regular day type. For example, '*Wednesdays, during term times for City of London School for Girls''.*

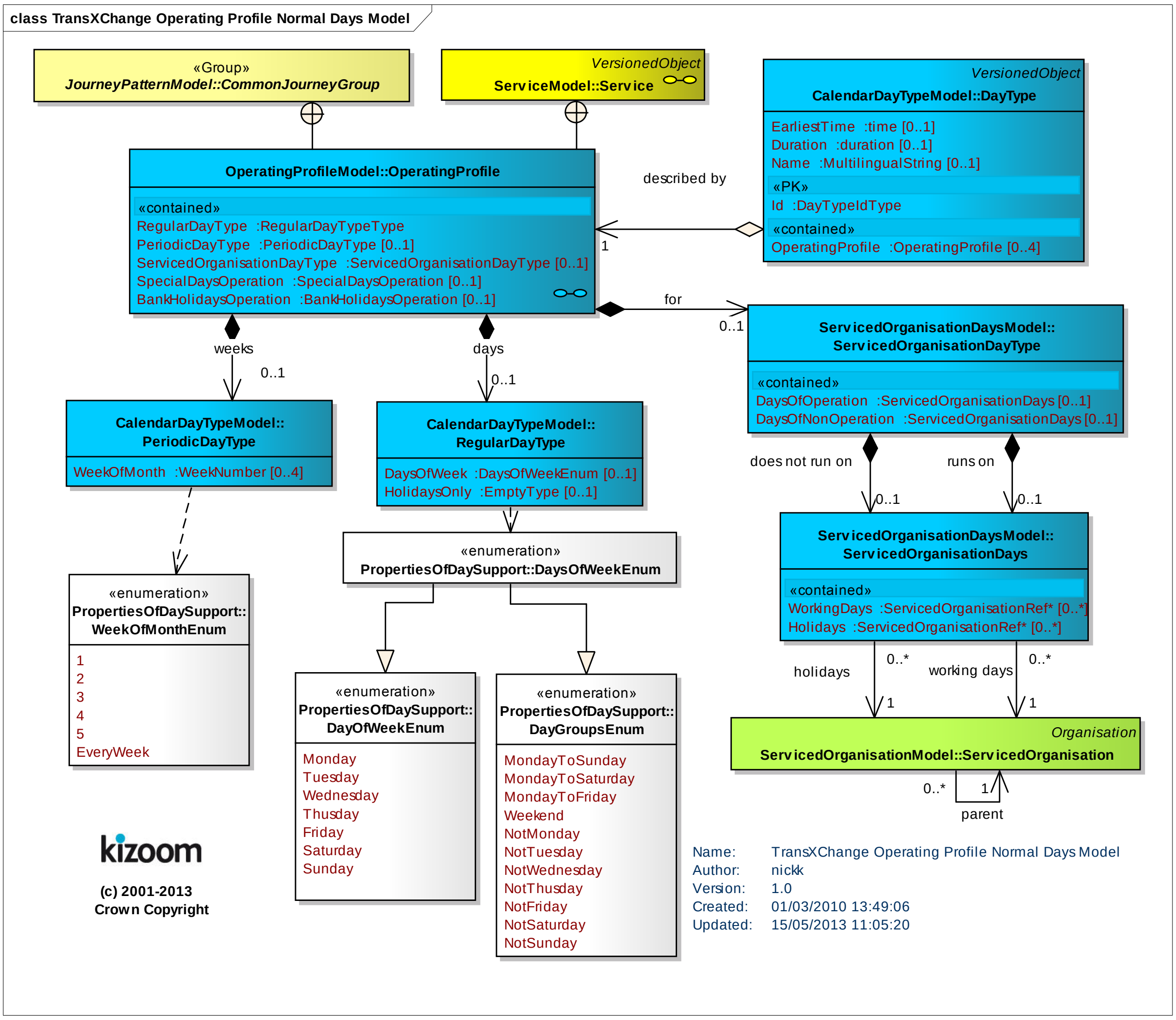


Figure 3-77 – UML Diagram of Normal Operation Profile

3.18.3 Exceptional Operation – OperatingProfile

The ***OperatingProfile / Special*** group specifies the exceptional operating days of a service. It can be made up of two distinct elements, as shown in the UML structure diagram in *Figure 3-78*:

- How the service operates on a bank holiday (***BankHolidaysOperation***). A number of different bank holiday day types are supported for both individual days and groups of days. For example '*Does not run Christmas, New Year's Day, Good Friday', 'Runs Bank Holiday Mondays'.* Day types include moveable feasts, such as *Easter Day*, whose day may vary from year to year. The holidays on which the service does (inclusion) or does not (exclusion) run are specified separately.
- Any special operating dates on which the service does (inclusion) or does not (exclusion) run (***SpecialDaysOperation***). Special days are always absolute calendar dates or calendar date ranges. For example '*does not run 11/11/2005'.* Special days override any Bank Holiday day types.
- **Special days** are intended just to record impending exceptions and so should be used sparingly and not be used to cover long periods of non-operation. They should not be used with operating period dates far in the past or the future.

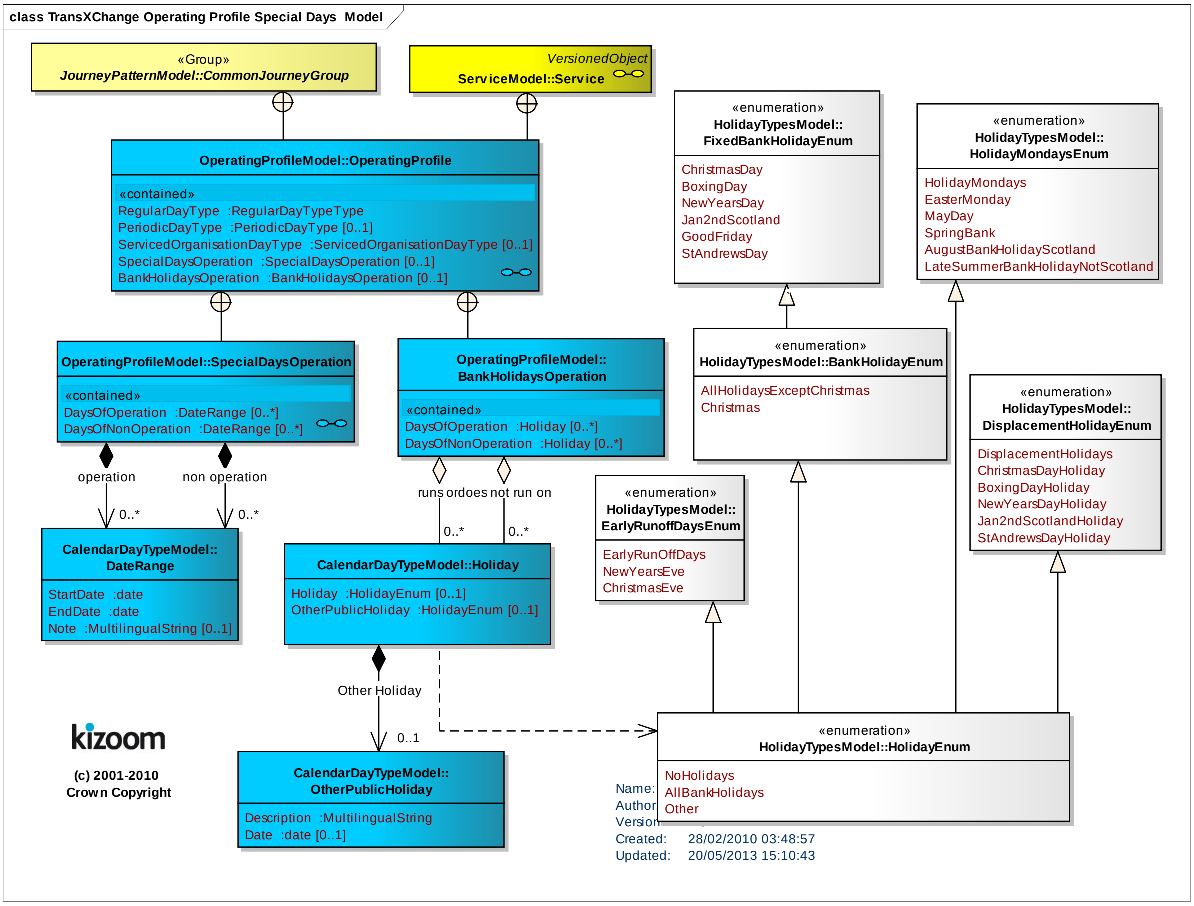


Figure 3-78 – UML Diagram of Special Operation Profile

A statement that a service does not operate on specific days should not be interpreted as implying that it operates on all other days. Similarly a statement that a service runs on a particular day does not necessarily imply that it does not run on all other days.

Note that the exclusion and inclusion of special days of operation have different meanings (see also 'General Principles for Using Operational Days' below):

- The Special Operation profile days of **non-operation** i.e. exclusion should be interpreted as further **constraining** the days of week and month of the Normal Operating Profile. For example, if the Normal Profile specifies that a service runs 'Monday to Friday', and the Special Operation Profile specifies that the Service does not run on New Year's Day, it will not run on New Year's Day, whatever day of the week New Year's Day occurs.
- The Special Operation profile days of explicit **operation** (i.e. inclusion) should be interpreted as being **additive** to the days of week and month of the Normal Operating Profile. For example, if

the Normal Operation Profile specifies that a service runs '*Sunday'*, and the Special Operation Profile specifies that the Service *does* run on New Year's Day, then the service also runs to that timetable on New Year's Day, regardless of the day of week New Year's Day falls.

Thus a typical usage is to have a lower frequency Service timetable that is used for Sundays and Bank Holidays, and a regular timetable that is used for weekdays, except when the weekday is a Bank holiday.

3.18.4 Services that Run for Specific ServicedOrganisation Working Days

Operational day types can be specified in terms of the working days or holidays of specified organisations, for example schools. A hierarchical parent relationship can be used to specify that working days are derived from those of another organisation, for example a Local Education Authority (LEAs), with specific variations.

3.18.4.1 ServicedOrganisation Model Introduction

Figure 3-79 introduces the ***ServicedOrganisation:***

- LEAs and their Schools are modelled in the schema by a ***ServicedOrganisation*** element. Each ***ServicedOrganisation*** may have a parent relationship (which should be acyclic) to another ***ServicedOrganisation***.
- Patterns of ***WorkingDays*** and ***Holidays*** may be specified for ***ServicedOrganisations***.
- Working days and holidays may be inherited from a parent organisation.
- Services and vehicle journeys may be associated with one or more organisations on the ***ServicedOrganisationDayType*** as part of the normal ***OperatingProfile***.

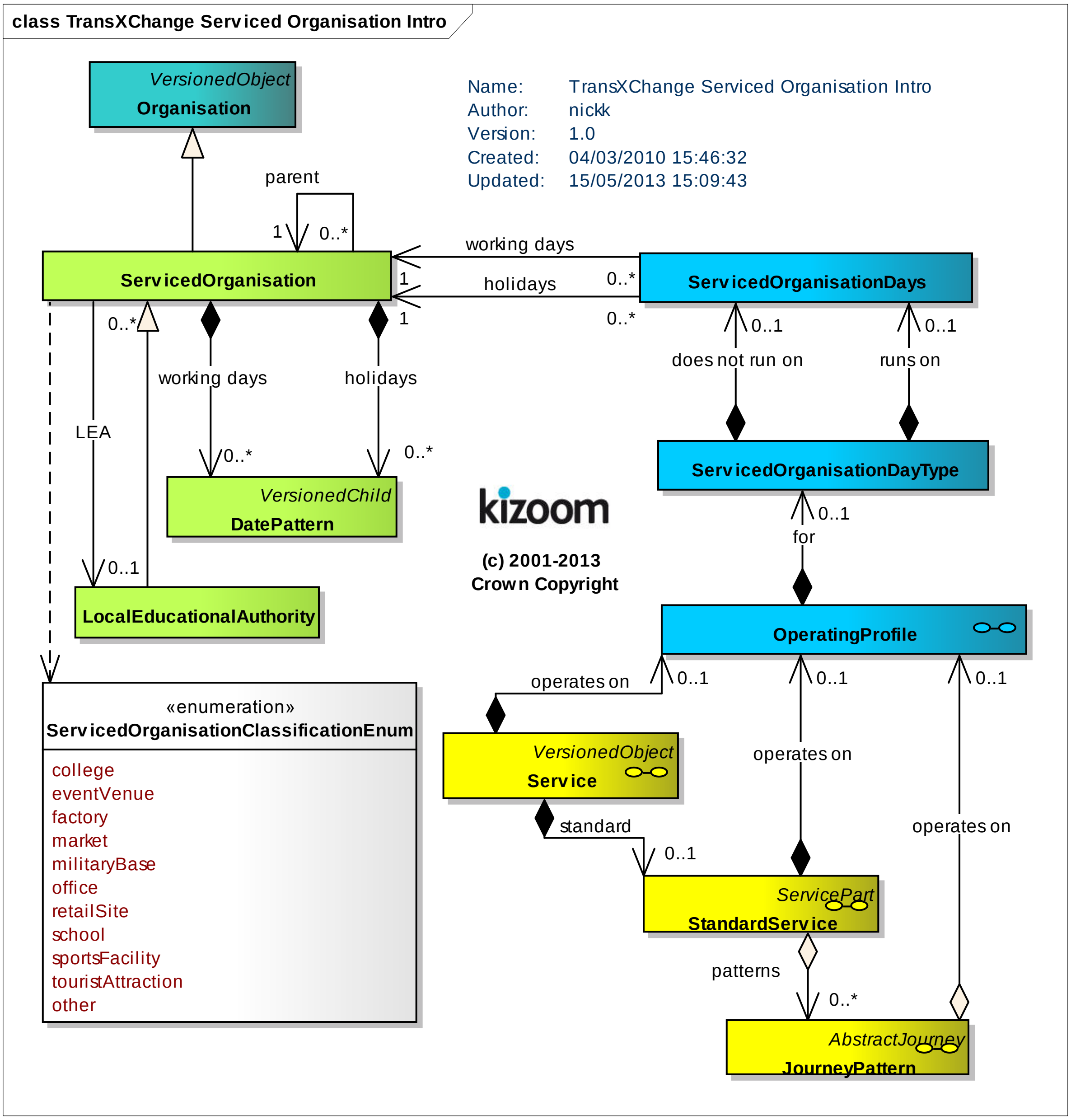


Figure 3-79 – UML Diagram of Serviced Organisation: Intro

3.18.4.2 ServicedOrganisation Model Details

Figure 3-80 shows further details of Serviced Organisations.

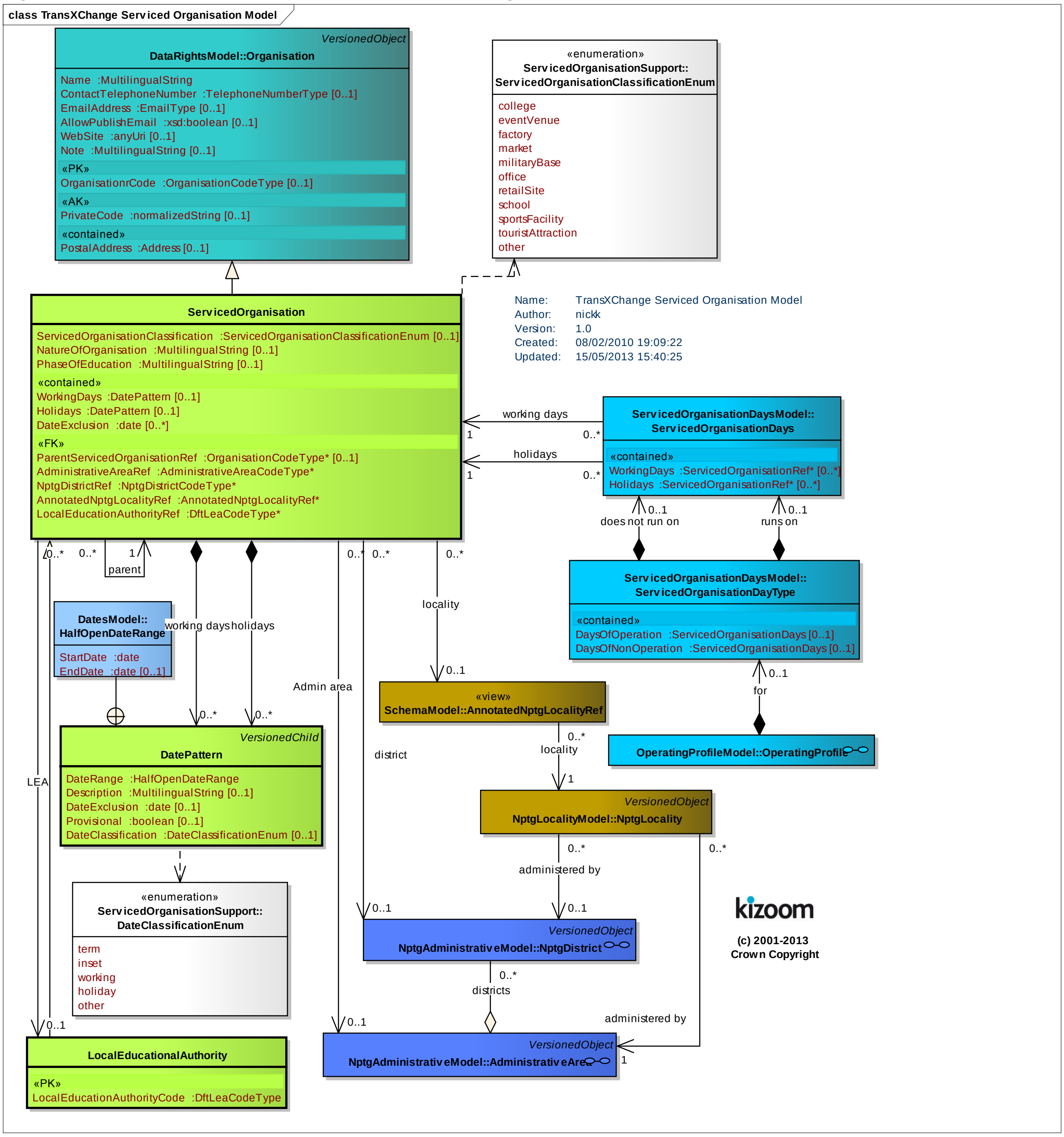


Figure 3-80 – UML Diagram of Serviced Organisation Days

3.18.5 General Principles for Using Operational Days

The *TransXChange* model has capabilities to specify operational days at a number of different levels of discourse (Service, Journey Pattern, Vehicle Journey, Journey Interchange); and to state operational days in both relative and absolute terms; that is,

(i) As general *day types*, such as 'Monday to Friday', or 'Christmas Day' (using the ***OperatingProfile / Normal*** elements).

(ii) As *absolute* calendar dates, such as '5th - 7th August 2005' (using the ***OperatingProfile / Special*** elements).

The different mechanisms can be combined to provide an overall set of operational conditions for a given vehicle journey that is to run on a given day of operation.

When interpreting a schedule, a number of simplifying rules are followed for combining the various element types to avoid ambiguity. The following general principles are followed for the use of operational days:

1. **Elements specified for a given profile property at a lower level of discourse completely replace the equivalent element at a higher level**. For example, if a Journey level operating profile specifies days of operation as '*Monday to Friday'* and a vehicle journey specifies '*Saturday'*, then the vehicle journey runs only on '*Saturday'*, not '*Monday to Saturda*y'. Similar considerations apply to Serviced Organisation operating days for parent and child Service Organisation levels.
2. **Lower level of discourse overrides higher level for operational days.** In particular, any operational days specified for a specific vehicle journey take precedence over those specified over the journey pattern; and any for the journey pattern over those specified for the whole service. For example, a vehicle journey may state more restricted or more extensive operation days than the overall service. *Table 3-17* shows the relative precedence of levels of discourse. Similarly Serviced Organisation properties override those of any parent organisation.

Operational days Precedence (1 high)	Level
1	***VehicleJourneyInterchange***.
2	***VehicleJourney.***
3	***JourneyPatternInterchange.***
4	***JourneyPattern***.
5	***Service***.

Table 3-17 – Precedence of Entity Levels

3. **Exceptional operation overrides regular operation at any given level**. Thus ***OperatingProfile*** **Special** dates override any dates indicated by ***OperatingProfile*** **Normal** day types.
4. **Exclusion constrains, inclusion adds.** Special days of non-operating further restrict the normal profile; special days of operation are additional to the profile.
5. **Non-operation overrides operation**. If *conflicting* overlapping dates for operation and non-operation days are specified, the non-operation is assumed to be correct at any given level of discourse. This applies only within each level of discourse – operational days at a lower level override i.e. replace all non-operational days at a higher level.
6. **More specific day type overrides less specific day type**. At any given level of discourse, more specific normal ***OperatingProfile*** day type values qualify the less specific values – as shown in *Table 3-18*.

Precedence (1 high)	Days
1	***ServicedOrganisationDayType*** days of non-operation.
2	***ServicedOrganisationDayType*** days of operation.
3	***PeriodicDayType*** qualifier.
4	***RegularDayType*** days.

Table 3-18 – Precedence of Normal Operation Day Types

3.18.6 Footnotes

The publisher generates explanatory footnotes from the day types, etc associated with each vehicle journey. These may arise from conditions specified at the Service, Journey Pattern or Vehicle Journey level. It may also append use defined notes.

3.18.6.1 Example - Footnotes

Figure 3-81 is an example of the automatically generated footnotes, output in order of generality.

Service operates from 22/09/2004 until further notice [P1]
Service runs Mondays on 3rd, 1st weeks in month [N1, N2]
Service runs during term times of Barham School [O2]
Service runs during holidays of Barham School [O1]
Service does not run during working days of Bootle Works 3 [O4]
Service does not run during holidays of Bootle Works [O3]
Service does not run Christmas Day, Boxing Day, New Year's Eve, Barday (20/12/2008) [H3, H4]
Service runs Bank Holiday Mondays, Good Friday, Fooday (12/12/2008) [H1, H2]
Service does not run 12/03/2008 until 12/04/2008 [S2]
Service runs 12/01/2008 until 12/02/2008 [S1]
Note: Electronic tickets accepted.

Figure 3-81 – Example of Footnotes

3.18.6.2 Footnote Variations

Table 3-19 shows the different footnotes generated for different TXC schema values (as updated for TXC v2.4).

			Argument	**Note/ Footnotes**	*Rule*
Period	**Range**		Start Date + End date	*“Service operates from [99/99/99] to [99/99/99]”*	*P1*
			Start Date + Open end date	*“Service operates from[99/99/99] until further notice “*	*P2*
Normal	**Days**	Regular Day Type	DayOfWeek	*“Service Operates [Monday, Tuesday, etc] only.”*	*N1*
	Weeks	Periodic Day Type	WeekNumber	*“Service runs Mondays, Thursdays on 3rd, 1st weeks in month”*	*N2*
	Holidays Only		HolidaysOnly	If specified at service level, amends heading to say: *“Holidays Only”* If at Vehicle Journey level adds footnote: *“Only runs on specified Public Holiday”*	*N3*
	Default	*No values*	*No values*	If there is no profile generate the default values: *“Service runs every day of the year including Christmas Day and Boxing Day.”*	*N0*
Serviced Organisation	**Days Of Operation**	Holidays	Serviced-Organisation/Id	*“Service runs only during holidays of [Serviced Organisation Name]”*	*O1*
		Workdays	Serviced-Organisation/Id	*“Service runs only during working days of [Serviced Organisation Name]*.” If The Organisation is a school should be: *“Service runs only during term time of [S o name]”*	*O2*
	Days Of non-Operation	Holidays	Serviced-Organisation/Id	*“Service does not run during holidays of [Serviced Organisation Name]*.”	*O3*
		Workdays	Serviced-Organisation/Id	*“Service does not run during working days of [Serviced Organisation Name].”* If the Organisation is a school should be *“Service does not run during term time of [Serviced Organisation Name] “*	*O4*
Exceptions	**Special Days**	Days Of Operation	DateRange + Note	*Service also runs [12/01/2008] until [12/02/2008]. Note.*	*S1*
		Days Of non-Operation	DateRange + Note	*Service does not [12/03/2008] until [12/04/2008]. Note.*	*S2*

	Holidays	Days Of Operation	BankHolidays, Displacement-Holidays, EarlyRunOffDays	*Service also runs [Christmas Day, Christmas Day Holiday, etc,]* Unless BankHolidayOnly *Service only runs [Christmas Day, Christmas Day Holiday], etc*	*H1*
			OtherPublicHoliday +Note	*"Service also runs [Description] (99/99/99)"* Unless BankHolidayOnly *"Service only runs [Description] (99/99/99)"*	*H2*
		Days Of non-Operation	BankHolidays, Displacement-Holidays, EarlyRunOffDays +Note	*"Service does not run Christmas Day, Christmas Day Holiday, etc "*	*H3*
			OtherPublicHoliday +Note	*"Service does not run Christmas Day, Christmas Day Holiday, etc"*	*H4*
Notes	Note		User defined text	*User defined text*	*H7*

Table 3-19 – Footnotes generated for different inputs

3.18.6.3 Serviced Organisation Calendar Example

The following example shows calendars for two levels of service organisation - an LEA which sets default term times and a School which sets some specific overrides. The results are summarised in A Calendar of holidays can be associated with each service organisation. For example

```
ServicedOrganisations>
<!- =====Parent LEA ======== ->
<ServicedOrganisation>
        <OrganisationCode>SO_1</OrganisationCode>
        <Name>Bleakshire LEA</Name>
        <WorkingDays>
                <DateRange>
                        <StartDate>2004-09-01</StartDate>
                        <EndDate>2004-12-23</EndDate>
                        <Description>Michaelmas Term</Description>
                        <DateClassification>term</DateClassification>
                </DateRange>
                <DateRange>
                        <StartDate>2005-01-01</StartDate>
                        <EndDate>2005-04-30</EndDate>
                        <Description>Easter Term</Description>
                        <DateClassification>term</DateClassification>
                </DateRange>
                <DateRange>
                        <StartDate>2005-04-02</StartDate>
                        <EndDate>2005-07-29</EndDate>
                        <Description>Summer Term</Description>
                        <DateClassification>term</DateClassification>
                </DateRange>
        </WorkingDays>
        <Holidays>
                <DateRange>
                        <StartDate>2004-11-11</StartDate>
                        <EndDate>2004-11-11</EndDate>
                        <Description>Inset day</Description>
                        <DateClassification>inset</DateClassification>
                </DateRange>
                </DateRange>
                        <StartDate>2004-11-01</StartDate>
                        <EndDate>2004-11-07</EndDate>
                        <Description>Autumn Half term</Description>
                        <DateClassification>holiday</DateClassification>
                </DateRange>
                <DateRange>
                        <StartDate>2005-02-07</StartDate>
                        <EndDate>2005-02-14</EndDate>
                        <Description>Spring Half term</Description>
                        <DateClassification>holiday</DateClassification>
                </DateRange>
                <DateRange>
                        <StartDate>2005-05-25</StartDate>
```

```
                    <EndDate>2005-06-01</EndDate>
                    <Description>summer Half term</Description>
                    <DateClassification>holiday</DateClassification>
               </DateRange>
     </Holidays>
     </ServicedOrganisation>
```

Figure 3-82 – Example LEA Calendar

```
     <!- =====Dotheboy Organisation == -->
<ServicedOrganisation>
     <OrganisationCode>SO_2</OrganisationCode>
     <Name>Dothboy's Academy</Name>
     <WorkingDays>
          <DateRange>
               <StartDate>2005-04-01</StartDate>
               <EndDate>2005-07-30</EndDate>
               <Description>Summer Term</Description>
          </DateRange>
     <Holidays>
          <DateRange>
               <StartDate>2004-11-12</StartDate>
               <EndDate>2004-11-12</EndDate>
               <Description>Inset day</Description>
               <DateClassification>inset</DateClassification>
          </DateRange>
     </Holidays>

      <ParentServicedOrganisationRef>SO_1</ParentServicedOrganisationRef>
</ServicedOrganisation>
</ServicedOrganisations>
```

Figure 3-83 – Example School Calendar

Table 3-20 & Table 3-21 show possible output

Serviced Organisation:	SO_1			Bleakshire LEA
		From	**To**	**Description**
Working Days	SO_1	2004-09-01	2004-12-23	Michaelmas Term
	SO_1	2005-01-01	2005-04-30	Easter Term
	SO_1	2005-04-02	2005-07-29	Summer Term
Holidays	SO_1	2004-11-11	2004-11-11	Inset day
	SO_1	2004-11-01	2004-11-07	Autumn Half term
	SO_1	2005-02-07	2005-02-14	Spring Half term
	SO_1	2005-05-25	2005-06-01	summer Half term

Table 3-20 – Example Serviced Organisation Particulars for LEA

Serviced Organisation	SO_2			Dothboy's Academy
Parent	SO_1			Bleakshire LEA
		From	**To**	**Description**
Working Days:	SO_1	2004-09-01	2004-12-23	Michaelmas Term
	SO_1	2005-01-01	2005-04-30	Easter Term
	SO_2	2005-04-01	2005-07-30	Summer Term
Holidays	SO_1	2004-11-12	2004-11-12	Inset day
	SO_1	2004-11-01	2004-11-07	Autumn Half term

	SO_1	2005-02-07	2005-02-14	Spring Half term
	SO_1	2005-05-25	2005-06-01	summer Half term

Table 3-21 – Example Serviced Organisation Particulars for School

3.18.7 Frequent Services

A *Frequent Service* is a service that meets the regulatory requirements for being classified as a frequent service, i.e. that runs to a frequency of every ten minutes or less in accordance with the Statutory Requirement, and that is to be formally registered as constituting a *Frequent Service,* as indicated by a ***FrequentService*** flag. Note that in addition, a minimum and maximum time gap between services operating as a frequency service can also be specified using the ***MinimumFrequency*** and ***MaximumFrequency*** elements.

Journeys which comprise a frequent service do not have to run at an absolutely regular frequency interval - they could be quite variable, such as every 2 - 7 minutes, as long as no service interval exceeds 10 minutes between consecutive journeys. The service should be described as running to its lowest available frequency e.g. '*Frequent service at least every 7 mins.*').

3.18.8 Frequency Based Services

Independently of whether the service is legally a *Frequent Service*, the *TransXChange* schema supports a *Frequency Based* Service definition*:* that is, a service that runs to a regular frequency, for example '*every 5 minutes' or 'every 15 minutes'*, rather than to a specific timetable (and which may or may not be a statutory *Frequent Service* – in which case it would be phrased *'Frequent service at least every 5 mins.'*).
The frequency pattern of a ***VehicleJourney*** is described by a ***Frequency*** element which holds elements giving the frequency of the service, and an end time. Frequencies may be specified either as an ***Interval*** of minutes (see *Table 3-23),* or as a collection of ***MinutesPastTheHour*** instances (see *Table 3-24)*.

The *TransXChange* schema allows the departure times for vehicle journeys occurring at regular intervals to be coded efficiently as a single vehicle journey, with a frequency to be repeated and an end-time (i.e. the last departure time that follows the standard pattern). Such journeys may or may not be part of a *Frequent Service*. Using the mechanism, just one vehicle journey is needed in the document rather than, say, many journeys that are identical but for the departure time. The interval is arbitrary – i.e. may be longer than that required to be a *Frequent Service* in the regulatory sense. The TransXChange Publisher will then generate the necessary

3.18.8.1 Frequency Described by Interval

*Table 3-22*shows a frequency based timetable described using a single journey and Frequency interval. Only the initial journey of a period of frequency based service need be explicitly given, so the entire timetable can be described by a single vehicle journey, as per column #1, together with a ***ScheduledFrequency*** (*15 minutes*) and an ***EndTime*** (*12:04*).

DepartureTime	*9:02*
ScheduledFrequency	*15*
EndTime	*12:02*
	#J1
Grub Street	9:02
Tin Pan Alley	9:12
Sinister Street	9:32

Table 3-22 – Frequency Service Timetable representation: Interval

Table 3-23 shows this as published - the Publisher generates the additional columns.

	J1	J1
	#1	**#2**
Grub Street	9:02	Then every 15 minutes until 12:02
Tin Pan Alley	9:12	
Sinister Street	9:32	

Table 3-23 – Example Frequent Service Timetable presentation: Interval

3.18.8.2 Departure Described by Minutes Past Hour

Table 3-24 shows an example of a service described using minutes past the hour. This can be used to describe services that don't run at regular intervals columns #1, #2 and #3 are all described by a single vehicle journey with a start and end time, and a frequency stated as two different minutes past the hour. Column #4 is a new journey.

Start time	*9:02*			
Minutes {Past Hour	--	*12*	*30*	**--**
EndTime				12:02
	#1	**#2**	**#3**	**#4**
		Then at the following minutes past the hour		**until**
Grub Street	9:02	12	30	12:02
Tin Pan Alley	9:12	22	40	12:12
Sinister Street	9:32	42	00	12:32

Table 3-24 – Example Frequent Service Timetable presentation: Minutes

3.18.8.3 Frequency Described on Multiple Individual Journeys – Merged Frequency

For some purposes it is useful to supply information about every single journey making up a *Frequent service*, for example, so as to be able to specify operational Block and Run information on journeys for AVL systems. Within the period of frequent operation these multiple, individually timed journeys can still be published as a single Frequency Group, that is a column of start times and a column giving the frequency, rather than separate columns for each journey. The TransXChange publisher will perform this merging of separate journeys as follows:

- If successive vehicle journeys are (a) flagged as *Frequency Based* and (b) have the same ***EndTime*** as the previous journey, then they will be collapsed into a single Frequency column.
- The indicated frequency values should normally also be the same for all journeys (***ScheduledFrequency, MinimumFrequency***, ***MaximumFrequency,).*** If they differ the values from the first journey will be used and a diagnostic will beaded to the validation report.
- *If any frequent services are provided as individual journeys for a frequent service in a document then all the individual journeys should be provided*.
- Note that the individual vehicle journeys themselves do not have to be at exactly regular intervals.
- The merging of journeys by the publisher can be suppressed using the mergeFrequencyJourneys option. (This is useful for those that wish to see the underlying data).

Thus one might have many journeys, the journey intervals are all slightly different as indicated by different start times, but less than the 10 minutes but rather than construing them as separate journeys in Table 3-25 below.

Departure time	*9:02*	9:10	9:16	9:21		12:02
ScheduledFrequency	*7*	*7*	*7*	*7*		*7*
MinimumFrequency	*6*	*6*	*6*	*6*		*6*
MaximumFrequency	*8*	*8*	*8*	*8*		*8*
EndTime	12:02	12:02	12:02	12:02		12:02
	J1	**J2**	**J3**	**J4**		**J(n)**
Grub Street	9:02	9:10	9:16	9:21	Etc	12:02
Tin Pan Alley	9:12	9:20	9:26	9:31		12:12
Sinister Street	9:32	9:40	9:36	9:51		12:32

Table 3-25 – Multi-journey Representation of Frequency Based journeys

The Publisher would present the journeys more concisely as in Table 3-26. (The actual text in column #2 will vary as per *Table 3-31*.

	j1	j2 to j(n-1)	j(n)

	#1	#2	#3
Grub Street	9:02	Frequent service at an interval of no more than **7** mins	12:02
Tin Pan Alley	9:12		12:12
Sinister Street	9:32		12:32

Table 3-26 – Merged presentation of separate Frequency journeys having the same frequencies and end time

3.18.8.4 Multi-journey to single group, multiple frequencies

If the frequency changes between journeys, as in *Table 3-27*.

Departure time	*9:02*	9:17	9:32	9:47		11:02	11:20		16:00
ScheduledFrequency	*15*	*15*	*15*	*15*	*15*	*15*	*20*	20	*20*
EndTime	*1102*	*1102*	*1102*	*1102*		*11:02*	*1600*		*16:00*
	J1	**J2**	**J3**	**J4**		**J(n)**	**J(m)**		**J(p)**
Grub Street	9:02	9:17	9:32	9:47	Etc	11:02	11:20	Etc	16:00
Tin Pan Alley	9:12	9:27	9:42	9:57		11:12	11:30		16:10
Sinister Street	9:32	9:47	10:02	10:17		11:32	11:50		16:30

Table 3-27 – Multi-journey Representation of Two Frequencies

The Publisher can add additional columns to describe the change in frequency as in *Table 3-28.* The additional column would be triggered by separate ***EndTime*** values (11:02, then 16:00), not by the separate ***ScheduledFrequency*** value, If the end time is the same, only a single column will be shown with the first scheduled frequency.

	j1	j2 to j(n-1)	j(n)	j(m)	j(m+1) to j(p-1)	j(p)
	#1	**#2**	**#3**	**#4**	**#5**	**#6**
Grub Street	9:02	Then every 15 minutes	11:02	11:20	Then every 20 minutes	16:00
Tin Pan Alley	9:12		11:12	12:32		16:10
Sinister Street	9:32		11:32	13:42		16:40

Table 3-28 – Merged presentation of separate Journeys with different frequencies

3.18.9 Partial Frequency Based Services (+TXC v2.4)

Some services are presented as being timetabled to absolute times at certain stops and as Frequency Based for other parts of the journey. This may be specified by an override Frequency ***Interval*** on the individual ***VehicleJourneyStopUsage*** of the ***VehicleJourneyTimingLink*** of just those stops that are to be shown as frequency based. This feature can only be used if journeys are individually specified (See Merged Frequency Journeys above).

Table 3-29 shows the representation of a three Partial Frequency Journeys:

Departure time	*9:02*	9:10	9:16	9:16
PartialFrequency	**false**	true	true	true
	J1	**J2**		**J3**
Grub Street				
Tin Pan Alley				
Hanger Lane		PT5	PT5	PT8
Rotten Row		PT5	PT5	PT6
Sinister Street				

Table 3-29 – Representation of Partial Frequency Based journeys

Table 3-30 shows the resulting presentation of the Partial Frequency Journeys in a matrix:

Departure time	*9:02*	9:10	9:16	9:21
	J1	**J2**	**J3**	**J4**
Grub Street	9:02	9:10	9:16	9:21
Tin Pan Alley	9:12	9:27	9:42	9:57
Hanger Lane	9:18	About every 5 minutes	About every 5 minutes	About every 8 minutes
Rotten Row	9.22			About every 6 minutes
Sinister Street	9:32	9:40	9:36	9:51

Table 3-30 – Presentation of Partial Frequency Based journeys

3.18.9.1 Text Descriptions for Frequency service

The text caption used in a column to describe the frequency is generated from the values of the ***ScheduledFrequency MinimumFrequency***, ***MaximumFrequency*** associated with the journey, as per *Table 3-31*. For registrations the less informative statutory definition is used.

Case	Frequent Service	Scheduled-Frequency Interval (mins)	Minimum-Frequency Interval (mins)	Maximum Frequency Interval (mins)	Result Phrase to show in matrix column for NON-REGISTRATION details	Result Phrase to show in matrix column for REGISTRATIONS
1	true	x	-	-	*then about every [**x**] minutes until*	*Frequent service at least every10 mins until*
2	true	x	m	-	*then at [**m-x**] minutes intervals until*	*Frequent service at least every10 mins until*
3	true	x	-	n	*then at intervals of no more than [**n**] mins until*	*Frequent service at least every 10 mins until*
4	true	x	m	n	*then at[**m-n**] minutes intervals until*	*Frequent service at least every 10 mins until*
5	false	x	-	-	*then about every [**x**] minutes until*	*Then every[**x**] mins until*
(6)	false	x	m	-	*then about every [m-x] minutes until*	*(Then every[x] mins until)*
(7)	false	x	-	n	*then at intervals of no more than [n] mins until*	*(Then every[x] mins until)*
(8)	false	x	m	n	*then at[m-n] minutes intervals until*	*(Then every[x] mins until)*

Table 3-31 – Frequency service Text Descriptions

3.18.10 Service Operational Days

Using the principles given above, we can summarise the use of the operational day elements shown in Figure 3-77 to specify the operational days of a ***Service*** as follows:

1. Each ***Service*** has an ***OperatingPeriod*** defining its overall start and end dates. All operational dates must fall within this period. The end date may be open.
    a. ***Operating period*.** The start date should be the first date of service operation or a date very close to it, if the start date is being aligned with other service changes. For example, it is in order for a Tuesday only service to have an operation period start on the preceding Sunday. A dummy date value e.g. 0000-00-00 or 1901-00-00 should not be used.
    b. **StartDate**: With the possible exception of certain London and Northern Ireland lines, no start date for a bus service in the UK should have a start date earlier than 26 October 1986.
    c. **EndDate**: The end date should be omitted if no end date is known, since unrealistic future dates such as year 9999, can create a misleading impression of certainty
    d.

2. Each ***Service*** can have a *default* ***OperatingProfile*** describing its operation: the ***Service*** / ***OperatingProfile*** normal elements are used to provide default values for all vehicle journeys of the service. Regular days can be specified in any or all of three ways (which can be combined together):
    a. ***RegularDayType***: Any combination of days of the week on which the service runs. Defaults to ***MondayToSunday*** if not specified.
    b. ***PeriodicDayType***: Additional qualifier of specific weeks of month on which the regular service runs.
    c. ***ServicedOrganisationDayType***: Dates defined by the working days or holidays of a named organisation, such as a school or Local Education Authority. ServicedOrganisations can also be used to represent other types of organisation such as *Works, Football Stadia.*

    Special days can be specified in two ways (which can be combined together):

d. ***BankHolidayOperation:*** Specific named ***BankHoliday*** day types (for example, ***ChristmasDay***; see ***BankHoliday*** element, or instances of one-off holidays (such as, say, a Silver Jubilee) described by an ***OtherPublicHoliday*** instance, are assigned to one of two categories:
   i. *Days of operation*. Bank holiday days for which the service operates. If the specified days of *operation* overlap with days of non-operation, the days of non-operation take precedence.
   ii. *Days of non-operation*. Bank holiday days for which the service does not operate. If the specified days of non-operation overlap with days of operation, the days of non-operation take precedence over days of operation.

e. ***SpecialDaysOfOperation:*** Specific ***DateRange*** elements, assigned to one of two categories:
   i. *Days of operation*. If the specified days of operation overlap with days of non-operation, the days of non-operation take precedence.
   ii. *Days of non-operation*. If the specified days of non-operation overlap with days of operation, the days of non-operation take precedence over days of operation.

3. Each ***JourneyPattern*** can also have a *specific* ***OperatingProfile,*** describing its individual operational days. The profile is made up of the same elements as the Service profile. Any values override the service level values.

4. Each ***VehicleJourney*** can have a *specific* ***OperatingProfile,*** describing its individual operational days. The profile is made up of the same elements as the Service profile. Any values override any service and journey pattern level values.

5. Interchange ***ValidityPeriod:*** As a further complication, a ***ValidityPeriod*** may be specified for individual interchanges at both the ***JourneyPatternInterchange*** and the ***VehicleJourneyInterchange*** level.
   a. Use of the Interchange is only valid during the specified validity period. The connection will not be available to the inbound vehicle journey except during the validity period.
   b. Any ***ValidityPeriod*** specified at the ***VehicleJourneyInterchange*** level overrides any ***ValidityPeriod*** specified at the ***JourneyPatternInterchange*** level.

Table 14-5 in Section 14.3; '*Precedence Rules for Dates'*, summarises the conditions for specifying dates.

3.18.11 Structure Example of Schedule with Operational Day Exceptions

For an Example of a service using complex dates and times see the *Interchange Example*.

3.19 Calendar Model

In order to relate Operating days to DayTypes a ***Calendar*** can be used. A Calendar (provides a list of days that can be assigned to specific named day types.

3.19.1.1 Calendar Model Introduction

Figure 3-84 introduces the ***Calendar*** Model. A ***Calendar*** is made up of one or more ***OperatingDays***. Each has a date and may be assigned to a day type (e.g. ***MondayToFriday***, ***Sunday***, etc)

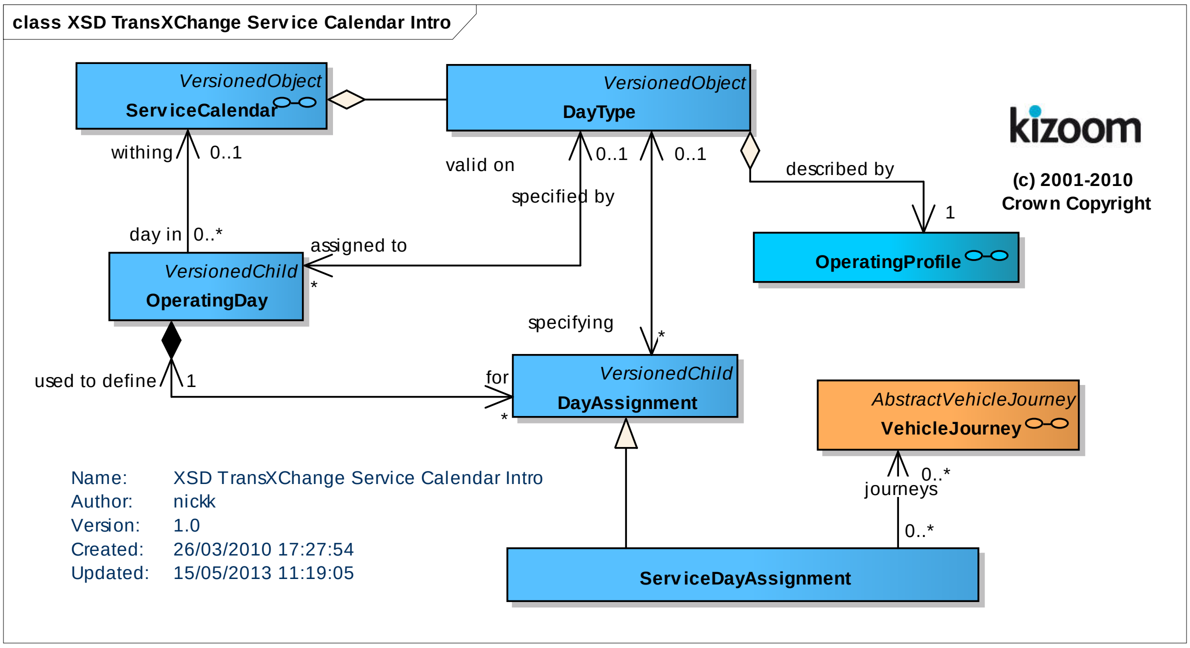


Figure 3-84 – UML Diagram of Calendar Model: Introduction

3.19.1.2 Calendar Model Details

Figure 3-85 shows further details of the ***Calendar*** Model

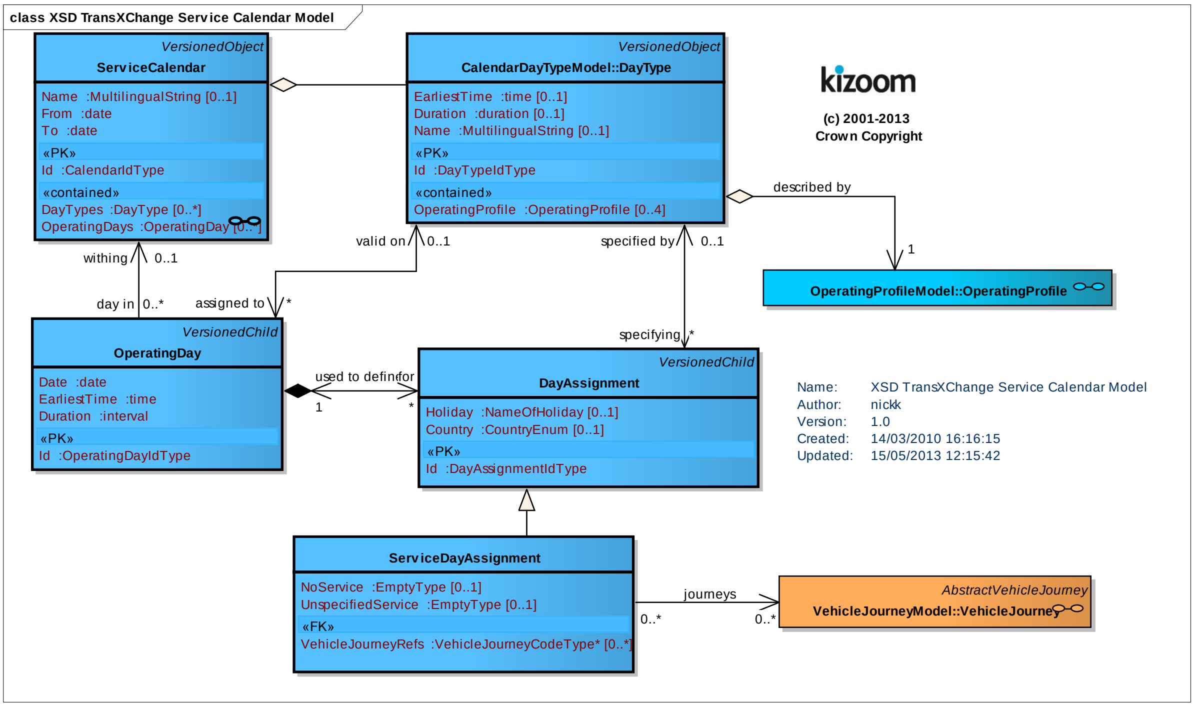


Figure 3-85 – UML Diagram of Calendar Model: Details

3.19.2 Calendar Model Hierarchy

Figure 3-86shows the inheritance hierarchy for the ***Calendar*** model elements.

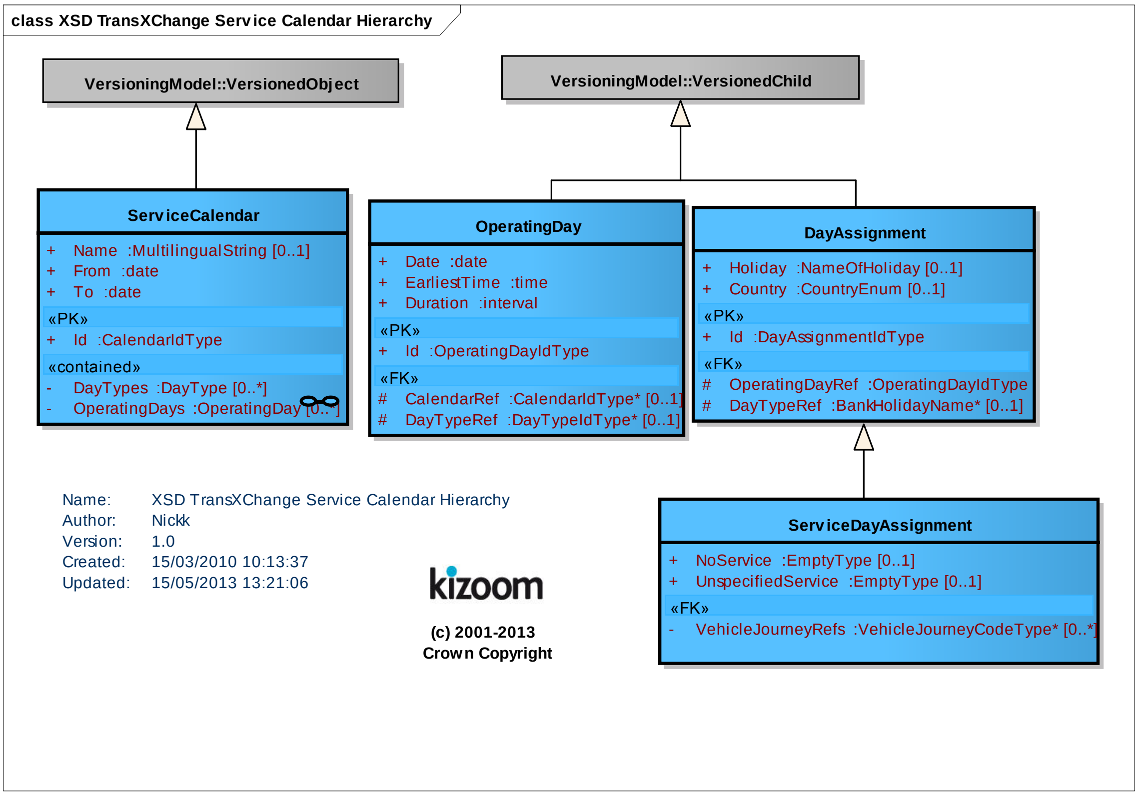


Figure 3-86 – UML Diagram of Calendar Element Hierarchy

3.20 Summary of TransXChange Entities and Identifiers

Table 3-32 summarises the significant entities of the *TransXChange* model. It also shows the identifiers used for each element and their scope (which in all cases must be unique within a document). The element identifiers fall into three scope groups:

- *External Codes forming part of well-defined national data systems* ('A'). For example the ***AtcoCode***, as defined in the *NaPTAN* data set. External codes are modelled as elements. These identifiers will always remain the same on repeated reissues of a given schedule as a *TransXChange* document.
- *External Codes forming part of arbitrary data systems.* ('B'). External codes are modelled as XML elements, and their names generally end in either '***Code'*** or '***Number'.*** These identifiers will normally remain the same on repeated reissues of a given schedule as a *TransXChange* document.
- *Internal Identifiers* used to identify objects locally within a document ('C'). Internal identifiers are modelled as an ***id*** attribute on the entity element. Most id attributes are of type ***IdType***. It is up to the application to decide whether internal identifiers should persist between different versions of a document. Typically there is no guarantee that these will remain the same on repeated reissues of a given schedule as a *TransXChange* document, though implementers are free to make them so if they wish.

Entity	Identifier				
	Type	Req-uired	Name	Has Private Code	Scope
StopPoint	Element	R	***AtcoCode,***	Yes	A-National
	Element	O	***NaptanCode***	Yes	A-National
StopArea (Cluster)r	Element	R	***StopAreaCode***	Yes	A-National
AdministrativeArea	Element	R	***AdministrativeAreaCode***	No	A-National
NptgLocality	Element	R	***NptgLocalityCode***	Yes	A-National
ServicedOrganisation	Element	R	***OrganisationCode***	TXC2.4	B-Various
Garage	Element	R	***GarageCode***	No	B-Operator
Service	Element	R	***ServiceCode,***	Yes	B-Operator
	Element	P	***TicketMachineServiceCode***		B-Operator
Operator	Element	O	***id***	Yes (TXC2.4)	C-Document
	Element	O	***NationalOperatorCode***		A-National
	Element	O	***OperatorCode***		B-Regional
	Element	O	***LicenceNumber***		A-National
Registration	Element	R	***VosaRegistrationNumber***	No	A-National
	Element	R	***/ TanCode***	No	A-National
	Element	R	***/ LicenceNumber***	No	A-National
	Element	R	***/ RegistrationNumber***	No	A-National
	Element	R	***VariationNumber***	No	B-Registration
Line	Element	R	***LineName***	No	B-Service
	Attribute	*O*	***id***		C-Document
Route	Attribute	O	***id***	Yes	C-Document
RouteSection	Attribute	*O*	***id***	No	C-Document
RouteLink	Attribute	*O*	***id***	No	C-Document
JourneyPattern	Attribute	*O*	***id***	Yes	C-Document
JourneyPatternSection	Attribute	*O*	***id***	No	C-Document
JourneyPatternTimingLink	Attribute	*O*	***id***	No	C-Document
JourneyPatternStopUsage	Attribute	O	***id***	No	C-Document
JourneyPatternInterchange	Attribute	O	***id***	No	C-Document
VehicleJourney	Element	*R*	***VehicleJourneyCode***	Yes	B-Service
	Element	*O*	***TicketMachineJourneyCode***		B-Service
VehicleJourneyTimingLink	Attribute	*O*	***id***	No	C-Document
VehicleJourneyStopUsage	Attribute	O	***id***	No	C-Document
VehicleJourneyInterchange	Attribute	O	***id***	No	C-Document
DeadRun	Attribute	*O*	***id***	No	C-Document
PositioningLink	Attribute	*O*	***id***	No	C-Document
LayoverPoint	Attribute	O	***id***	No	C-Document
Location	Attribute	O	***id***	No	C-Document
AccessVehicleEquipment	Element	O	***id***	TXC2.5	C-Document
AssistanceService	Element	O	***id***	TXC2.5	C-Document
ServiceFacilitySet	Element	O	***id***	TXC2.5	C-Document
Calendar	Element	O	***id***	TXC2.4	C-Document
Contributor	Element	O	***id***	TXC2.4	C-Document

DataRight	Element	O	***id***	TXC2.4	C-Document
SupportingDocument	Element	*R*	***DocumentUri***	No	C-Document
Note	Element	*R*	***NoteCode***	No	B-Service

Table 3-32 – Main Entities of the TransXChange Model

The uniqueness scope of many identifiers is formally defined in the *TransXChange* schema by XML **keyref** constraints. See 'Integrity Rules' in *Section 14.*

3.20.1 Summary of TransXChange Entity Identifier Types

Table 3-33 shows the data types for the main identifiers used in the *TransXChange* model. ***(Changed in +TXC 2.4***).

Name	Base Type (2.4)	Former Type (TXC 2.2)	Restrictions
id	Xsd:NMTOKEN	Xsd:NMTOKEN	
AtcoCodeType	Xsd:normalizedString	Xsd:string	Yes
NaptanCodeType	Xsd:normalizedString	Xsd:string	Yes: 1 or alpha8 p[prefix
StopAreaCodeType	**Xsd:normalizedString**	Xsd:string	
AdministrativeAreaCodeType	Xsd:NMTOKEN	Xsd:NMTOKEN	Restricted Code set
AtcoAreaCodeType	Xsd:NMTOKEN	Xsd:NMTOKEN	
NptgLocalityCodeType	Xsd:NMTOKEN	Xsd:NMTOKEN	Restricted Prefix
OrganisationCodeType	Xsd:NMTOKEN	Xsd:NMTOKEN	
GarageCodeType	Xsd:NMTOKEN	Xsd:NMTOKEN	
ServiceCodeType	**Xsd:normalizedString**	Xsd:NMTOKEN	
TicketMachineServiceCodeType	**Xsd:normalizedString**	Xsd:NMTOKEN	
TicketMachineJourneyCodeType	Xsd:NMTOKEN	Xsd:NMTOKEN	
NationalOperatorCode	Xsd:NMTOKEN	Xsd:NMTOKEN	
OperatorCode	Xsd:NMTOKEN	Xsd:NMTOKEN	
LicenceNumber	**Xsd:normalizedString**	Xsd:string	
VosaRegistrationNumber	na	na	
/ TanPrefixEnumeration	Xsd:string	Xsd:string	Enumerated value
/ OperatorPartialLicenceNumberType	**Xsd:normalizedString**	Xsd:string	
/ RegistrationNumber	**Xsd:normalizedString**	Xsd:string	
RegistrationVariationNumberType	Xsd:nonNegativeInteger	Xsd:integer	
LineName	Xsd:string	Xsd:string	
VehicleJourneyCode	Xsd:NMTOKEN	Xsd:NMTOKEN	
DocumentUri	anyUri	anyUri	

Table 3-33 – TransXChange Main Entity Identifier Types

3.20.2 Summary of NPTG Entity Identifier Types

Table 3-34 shows the data types for the NPTG identifiers used in the *TransXChange* model though the incorporation of NaPTAN Stop Points and Stop Areas ***(+TXC 2.4***).

Name	Base Type (2.4)	Type (2.2)	Restrictions
RegionCodeType	Xsd:NMTOKEN	Xsd:NMTOKEN	
RegionShortCodeType	Xsd:NMTOKEN	Xsd:NMTOKEN	Yes
AtcoAreaCodeType	Xsd:NMTOKEN	Xsd:NMTOKEN	Restricted Code set
NptgDistrictCodeType	Xsd:NMTOKEN	Xsd:NMTOKEN	
NaptanPrefixType	Xsd:NMTOKEN	Xsd:NMTOKEN	Restricted Code set
PlusbusZoneCodeType	Xsd:NMTOKEN	Xsd:NMTOKEN	
CallCentreCodeType	Xsd:NMTOKEN	Xsd:NMTOKEN	
NptgLocalityCodeType	Xsd:NMTOKEN	Xsd:NMTOKEN	Restricted Prefix

Table 3-34 – NPTG Main Entity Identifier Types

3.20.3 Private codes

For a number of semantically significant elements, an additional ***PrivateCode*** element is supported. The ***PrivateCode*** facilitates the general purpose exchange of data in *TransXChange* format, as it allows instances to be annotated with the alternative identifier, so as to allow the unambiguous reconciliation of element identity between different computer systems on round trip exchanges. *Table 3-32* also indicates the elements that can have a ***PrivateCode.***

Note: Private codes are used in preference to XML ANY element types, as the latter cause a reduction in the efficacy of some commonly used validators.

3.20.4 Referencing Elements

A systematic convention is used to show the implementation of relationships (other than inline containment) between elements. For each entity that is referenced, a ***RefStructure*** is defined (based on the same type as the identifier of the referenced element), and this structure is used to type all references. This helps when reading the schema – if you see an element with REF on it, you know it implements a relationship with another entity. *Table 3-35* lists the elements that are referenced in various relationships.

Entity	Reference	Type	Scope	Version
StopPoint	***StopPointRef***	***AtcoCodeType***	NaPTAN	2.1
StopArea	***StopAreaRef***	***StopAreaCodeType***	NaPTAN	2.1
AdministrativeArea	***AdministrativeAreaRef***	***AdminAreaCodeType***	NPTG	2.1
NptgLocality	***NptgLocalityRef***	***NptgLocalityCodeType***	NPTG	2.1
ServicedOrganisation	***ServicedOrganisationRef***	***OrganisationCodeType***	TXC	2.1
Contributor	***OrganisationRef***	***OrganisationCodeType***	TXC	2.4
Service	***ServiceRef***	***ServiceCodeType***	TXC	2.1
Operator	***OperatorRef***	***idType***	TXC	2.1
Registration	***RegistrationRef***	***VosaRegistrationNumber Structure***	TXC	2.1
Line	***LineRef***	***idType***	TXC	2.1
Route	***RouteRef***	***idType***	TXC	2.1
RouteSection	***RouteSectionRef***	***idType***	TXC	2.1
RouteLink	***RouteLinkRef***	***idType***	TXC	2.1
JourneyPattern	***JourneyPatternRef***	***idType***	TXC	2.1
JourneyPatternSection	***JourneyPatternSectionRef***	***idType***	TXC	2.1
JourneyPatternTimingLink	***JourneyPatternTimingLinkRef***	***idType***	TXC	2.1
JourneyPatternStopUsage	***JourneyPatternStopUsageRef***	***idType***	TXC	2.1
JourneyPatternInterchange	***JourneyPatternInterchangeRef***	***idType***	TXC	2.1
VehicleJourney	***VehicleJourneyRef***	***VehicleJourneyCode***	TXC	2.1
VehicleJourneyTimingLink	***VehicleJourneyTimingLinkRef***	***idType***	TXC	2.1
VehicleJourneyStopUsage	***VehicleJourneyStopUsageRef***	***idType***	TXC	2.1
VehicleJourneyInterchange	***VehicleJourneyInterchangeRef***	***idType***	TXC	2.1
DeadRun	***DeadRunRef***	***idType***	TXC	2.1
PositioningLink	***PositioningLinkRef***	***idType***	TXC	2.1
LayoverPoint	***LayoverPointRef***	***idType***	TXC	2.1
Location	***LocationRef***	***idType***	TXC	2.1
Calendar	***CalendarRef***	***idType***	TXC	2.4
DataRight	***DataRightRef***	***idType***	TXC	2.4
SupportingDocument	**--**	***DocumentUri***	TXC	2.1
Note	**--**	***NoteCode***	TXC	2.1

Table 3-35 – References to Entities in the TransXChange Model

3.20.5 Summary of TransXChange Ancillary Identifiers

Table 3-32 summarises the ancillary external entities referenced by the *TransXChange* model. It also shows the identifiers used for each element. The scope is local to the document in all cases.

Entity	Identifier	Base Type 2.4	Base Type 2.1
Stop/Plate	***PlateCodeType***	Xsd:normalizedString	Xsd:normalizedString
Stop/Cleardown	***CleardownCodeType***	Xsd:positiveInteger	Xsd:positiveInteger
DutyCrew	***DutyCrewCodeType,***	Xsd:NMTOKEN	Xsd:NMTOKEN
TicketMachineJourney	***TicketMachineJourneyCodeType***	Xsd:NMTOKEN	Xsd:NMTOKEN
TicketMachineService	***TicketMachineServiceCodeType***	Xsd:NMTOKEN	Xsd:NMTOKEN
FareStage	***FareStageNumberType***	Xsd:nonNegativeInteger	Xsd:nonNegativeIntege

			r
VehicleType	***NptgLocalityCodeType***	Xsd:NMTOKEN	Xsd:NMTOKEN

Table 3-36 – Externally referenced Entities of the TransXChange Model

3.21 Common Data types

3.21.1 XML Base Types

Figure 3-87 summarises the base XML data types used in NaPTAN & TransXChange.

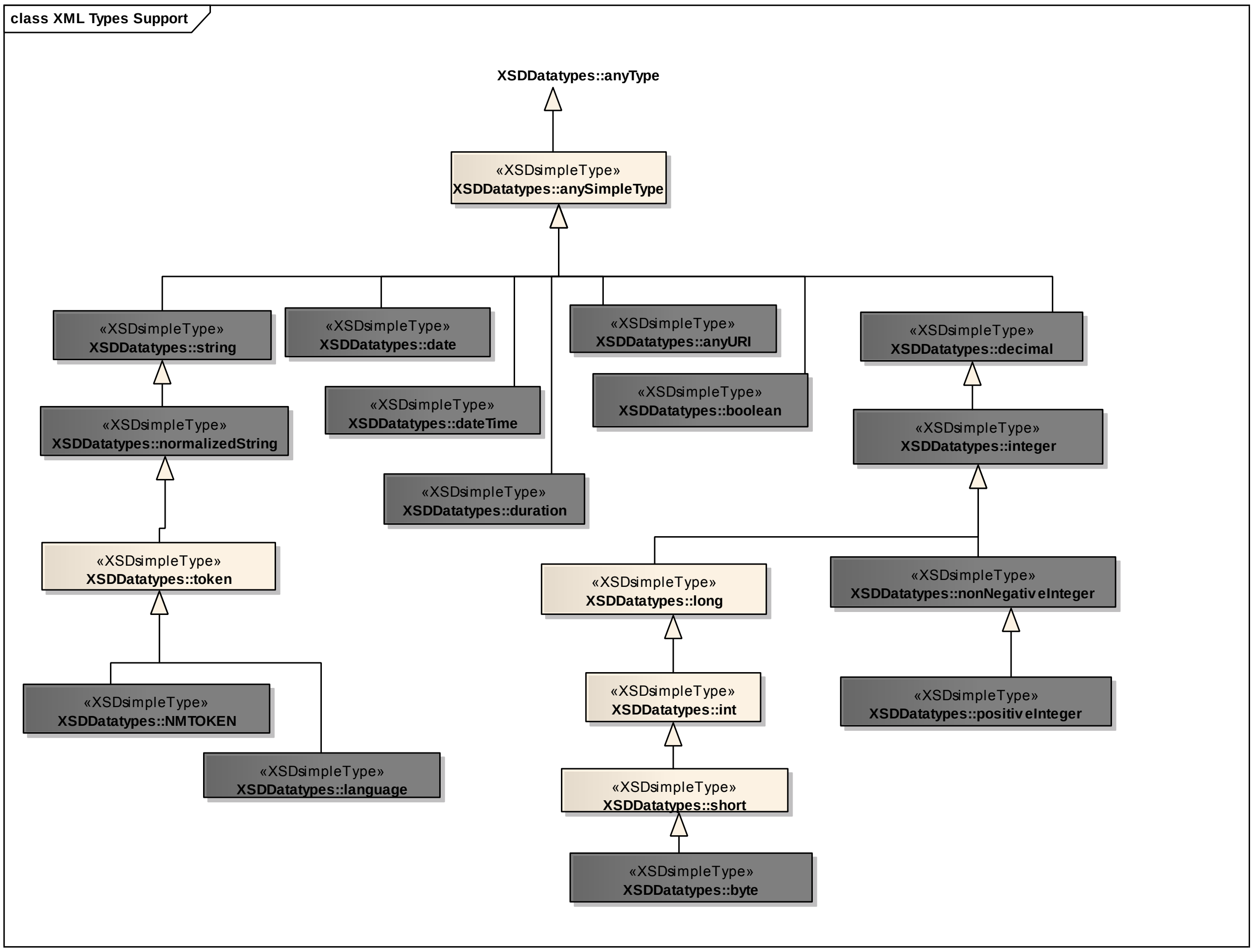


Figure 3-87 – UML Diagram of XML base Data types

3.21.2 General Utility Types

Figure 3-88 summarises the common utility XML data types used in NaPTAN & TransXChange.

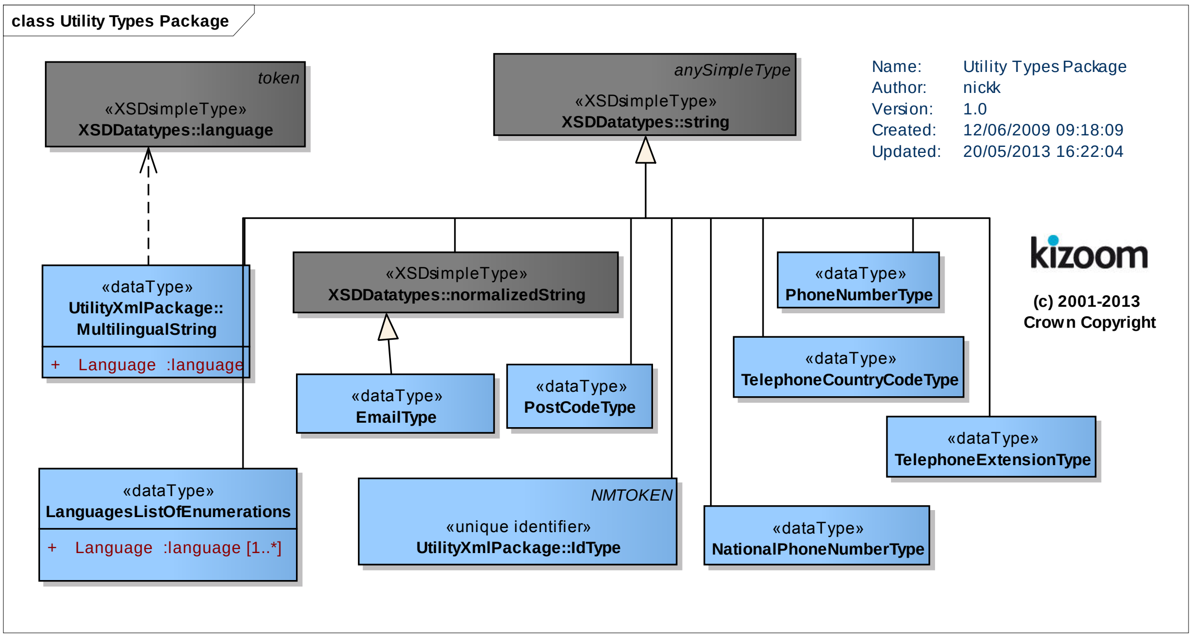


Figure 3-88 – Additional utility types used by TransXChange & NaPTAN

3.22 Data Rights Model

A *TransXChange* document can specify the allowed use of the data.

3.22.1 Data Rights Model - Introduction

Figure 3-89 introduces the rights model: a ***DataRight*** can be associated with every ***VersionedObject***. Data Rights may be held by an ***Operator***, or by a ***Contributor*** Organisation.

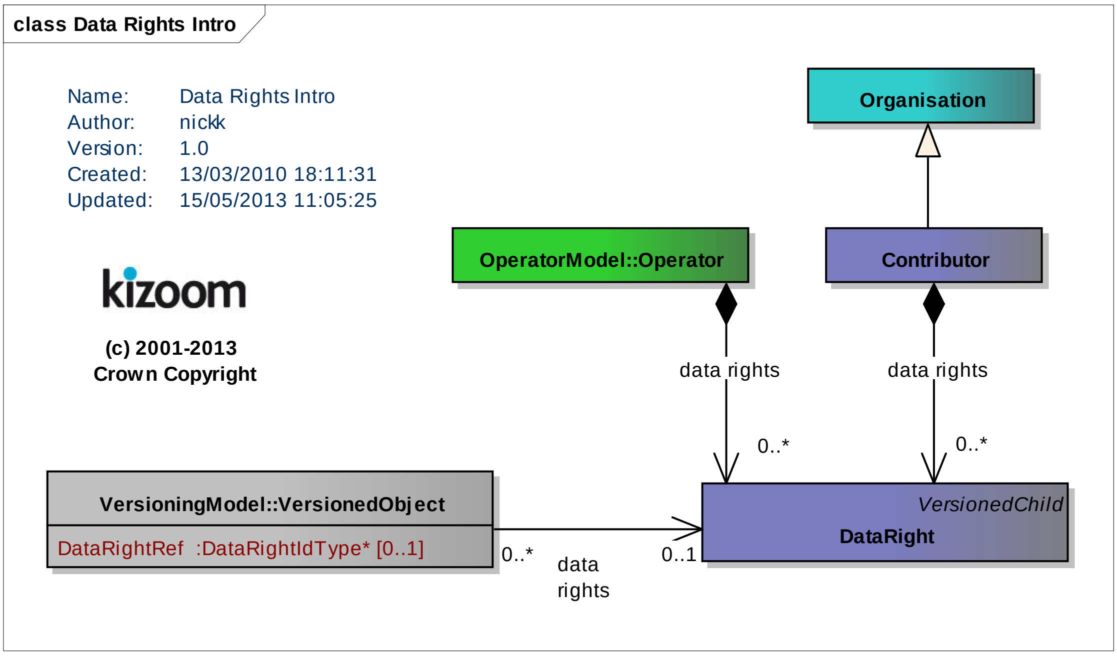


Figure 3-89 – UML Diagram of Rights Model: Introduction

3.22.2 Data Rights Model - Introduction

Figure 3-89 shows further details of the ***DataRight*** model: It is possible to specify **Copyright**, **Terms of Use** and a **Data Policy** – including whether the data is subject to Freedom of Information act, with each element. The ***DataRight*** of a parent element is deemed to apply to its children

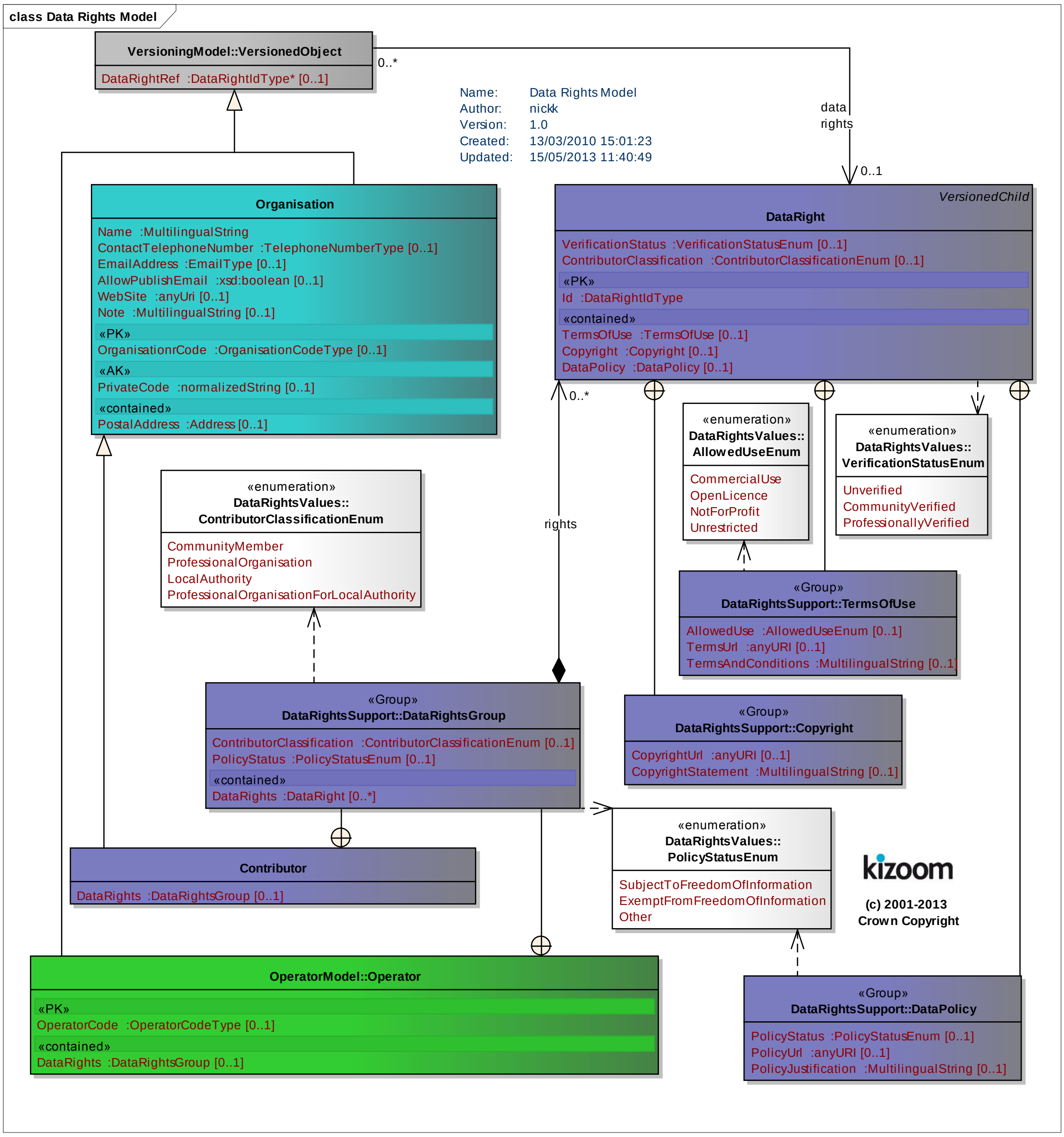


Figure 3-90 – UML Diagram of Rights Model: Details

4 WORKED EXAMPLE OF A TRANSXCHANGE SCHEDULE

This section provides a basic introductory example of using the main *TransXChange* elements. For more complex examples, refer to Section 5.

4.1 Worked Example: Bus Timetable

The elements of a *TransXChange* ***StandardService*** are illustrated using the fictional timetable shown in *Table 4-1*:

Service:	1, Suborn - Beall	Line:	1A	1B	1A	1C	1A	1B	1A
Origin:	Suborn, Bus Station								
Destination:	Beall, Bus Exchange								
Notes: Valid from 5 February until further notice									
Mondays to Fridays / Service on Mayday		Operator:	ACO	ACO	ACO	RED	ACO	ACO	ACO
		Journey:	37	38	39	40	41	42	43
Suborn, Bus Station		Depart:	15:55	16:15	16:35	16:40	16:55	17:15	17:35
Garden Village, Shop						16:46			
Robridge, Plough				16:26					
Barford, Red Lion		Arrive:	16:08	16:28	16:48	16:53	17:08	17:28	17:48
Barford, Red Lion		Depart:	16:09	16:29	16:49	16:54	17:09	17:29	17:49
Egham, Golden Lion			16:12		16:52	16:57	17:12		17:52
Godhill Church			16:15		16:55	17:00	17:15		17:55
Beall, Exchange			16:32	16:52	17:12		17:32	17:52	18:12
Beall Business Park, Shell				16:53		17:16		17:53	
Beall, Bus Station		Arrive:				17:17			

Table 4-1 – Worked Example: Bus Timetable

4.2 Worked Example: Service Components

To encode the example, we use a ***StandardService*** comprising:

- A ***LicensedOperator,*** who registers the service.
- A ***Registration,*** recording the statutory registration of the service with a TAN.
- A ***StandardService,*** recording the schedule of a fixed route service.
- A ***Route,*** over which the service runs from ***StopPoint*** to ***StopPoint***.
- A ***JourneyPattern,*** describing a general journey as a sequence of timing links.
- A collection of ***VehicleJourney*** instances, describing individual journeys as timing link sequences, and the departure times at which they run. Each ***VehicleJourney*** is based on a ***JourneyPattern*****.**

4.3 Worked Example: Operator

Two types of operator can be defined for a service: the ***RegisteredOperator*** who registers the service, and one or more ***AssociatedOperator*** instances, who may perform subsidiary roles.

In the example case the registering ***LicensedOperator*** is '*ACO'*, and the single associated ***Operator*** is '*RED'*, who runs one particular journey on behalf of 'ACO'.

4.4 Worked Example: Registration

The ***Registration*** holds administrative details of the service registration, such as ***SubmissionDate***, ***SubmissionAuthor,*** and the ***TrafficAreas*** with full or partial responsibility for the registration of the submission.

4.5 Worked Example: StopPoints

A NaPTAN ***AnnotatedStopPoint*** instance is used to reference each of the vehicle stops where passengers may embark or disembark. Each of these identifies a NaPTAN point.

In the example there are nine stops, each with a specified type and sub type. See *Table 4-2.*

			Stop Point			
Sequence	**StopPoint / Name**	**StopPoint / Name**	**AtcoCode**	**NaPTAN Code (SMS number)**	**Stop Type**	**Sub Type**
#1	*Bus Station*	*Suborn*	*0600000001*	*chsadad*	*BCS*	*MKD*
#2	*Shops*	*Garden Village*	*0600000002*	*chsadag*	*BCT*	*MKD*
#3	*Plough*	*Robridge*	*0600000003*	*chsadaj*	*BCT*	*MKD*
#4	*Red Lion*	*Barford*	*0600000004*	*chsadam*	*BCT*	*MKD*
#5	*Golden Lion*	*Egham*	*0600000005*	*chsadap*	*BCT*	*MKD*
#6	*Church*	*Godhill*	*0600000006*	*chsadat*	*BCT*	*CUS*
#7	*Exchange*	*Beall*	*0600000007*	*chsadaw*	*BCT*	*MKD*
#8	*Shell*	*Beall*	*0600000008*	*chsadga*	*BCT*	*MKD*
#9	*Bus Station*	*Beall*	*0600000009*	*chsadgd*	*BCS*	*MKD*

Table 4-2 – Worked Example: StopPoint Instances

4.6 Worked Example: Route and Tracks

A ***Route*** describes the sequence of stop points of the route, and contains an ordered collection of references to ***RouteSection*** elements. Each ***RouteSection*** comprises an ordered collection of ***RouteLink*** elements, making up the detailed stop sequence of the route. Links always run from *NaPTAN* ***StopPoint*** to *NaPTAN* ***StopPoint***. The spatial path of each ***RouteLink*** is described by one or more ***Track*** elements, each having a ***Mapping***; a collection of points (***Location*** elements) giving the physical path of the route between stops.

Figure 4-1 shows an example route: The route links all have tracks comprising several location points.

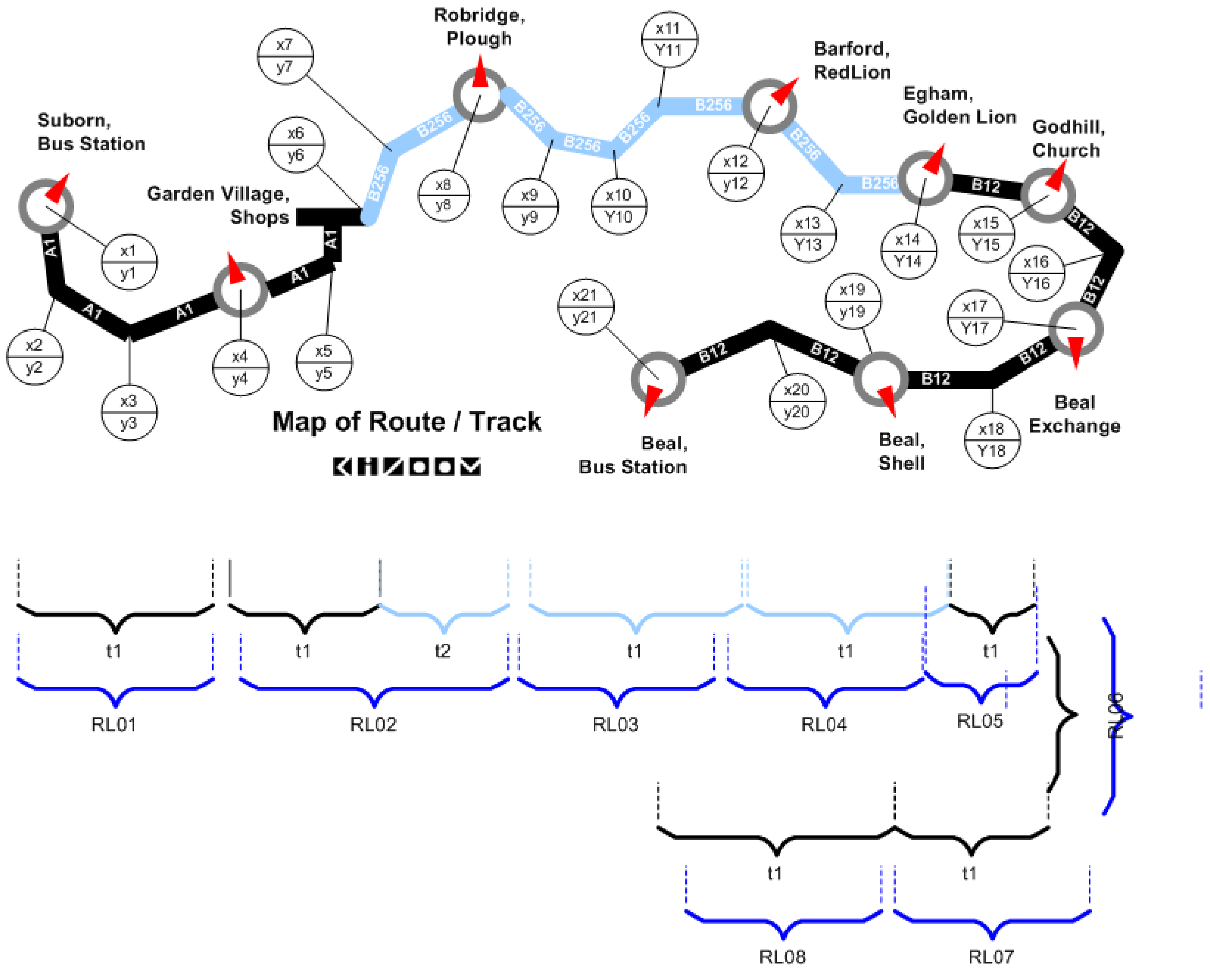


Figure 4-1 – Worked Example: Map of the Route

A single ***RouteSection***, with a link sequence of eight ***RouteLink*** instances (RL_01 – RL_08) suffices to connect the nine stops of the example, see *Table 4-3*. Each of the eight route links has a single ***Track***, except for the link between '*Garden Village'* and '*Robridge, Plough'*, which runs on two different roads (A1, B256), and so requires two track instance *(T_2_1, T_2_2).* The tracks have a varying number of intermediate points, depending on their spatial depiction.

Link	Track	Origin	Destination	Mapping	Map Ref	Distance	Bearing
RL_01	*T_1_1*	*Suborn, Bus Station*	*Garden Village, Shops*	*(x_1,y_1) (x_2,y_2) (x_3,y_3) (x_4,y_4)*	*A1*	*5573*	*E*
RL_02	*T_2_1*	*Garden Village, Shops*	(junction)	*(x_4,y_4) (x_5,y_5) (x_6,y_6)*	*A1*	*4512*	*NE*
	T_2_2	(junction)	*Robridge, Plough*	*$x_6,y_6)$ (x_7,y_7) (x_8,y_8)*	*B256*		
RL_03	*T_3_1*	*Robridge, Plough*	*Barford, Red Lion*	*(x_8,y_8) (x_9,y_9) (x_{10},y_{10}) (x_{11},y_{11}) (x_{12},y_{12})*	*B256*	*6046*	*E*
RL_04	*T_4_1*	*Barford, Red Lion*	*Egham, Golden Lion*	*(x_{12},y_{12}) (x_{13},y_{13}) (x_{14},y_{14})*	*B256*	*2520*	*NE*
RL_05	*T_5_1*	*Egham, Golden Lion*	*Godhill, Church*	*(x_{14},y_{14}) (x_{15},y_{15})*	*B12*	*1955*	*SE*
RL_06	*T_6_1*	*Godhill, Church*	*Beall, Exchange*	*(x_{15},y_{15}) (x_{16},y_{16}) (x_{17},y_{17})*	*B12*	*2963*	*SW*
RL_07	*T_7_1*	*Beall, Exchange*	*Beall, Shell*	*(x_{17},y_{17}) (x_{18},y_{18}) (x_{19},y_{19})*	*B12*	*3560*	*SW*
RL_08	*T_8_1*	*Beall, Shell*	*Beall, Bus Station*	*(x_{19},y_{19}) (x_{20},y_{20}) (x_{21},y_{21})*	*B12*	*2005*	*SW*

Table 4-3 – Worked Example: RouteLink Instances

4.7 Worked Example: JourneyPattern

A ***JourneyPattern*** represents the pattern of working for vehicles of the service, and is composed of an ordered collection of ***JourneyPatternSection*** instances, each containing an ordered collection of ***JourneyPatternTimingLink*** instances, together defining a specific sequence of timing links.

To model the bus service in the worked example, one ***JourneyPattern*** instance is used that defines an overall sequence of stops.

- Three main vehicle journey instances *(37, 38* and *40*) are defined that reference the journey pattern; each vehicle journey omits particular stops (see *Figure 4-2*).
- Four other vehicle journey instances (*39, 41, 42* and *43*) are exact replicas of the first three vehicle journeys, apart from a different departure time, and so can be defined simply by referencing the links of the appropriate journey.

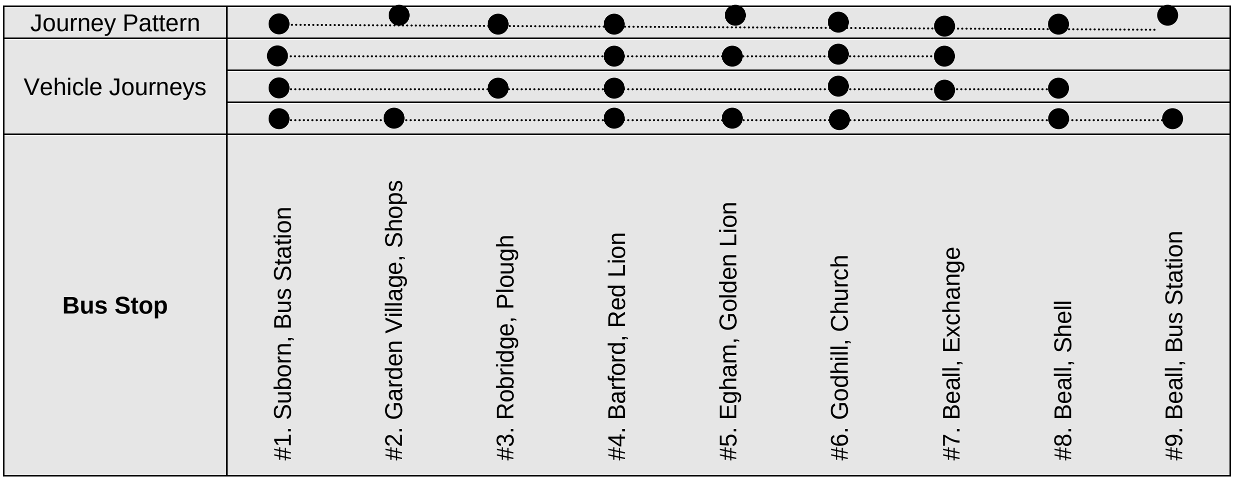


Figure 4-2 – Worked Example: Journey Pattern

The exact sequence of stops is given by a ***JourneyPattern.*** The journey pattern will also specify information about the use of the stop (which may vary according to service), in particular: The ***JourneyPatternTimingLink*** instances for the example journey pattern are shown in *Table 4-4.*

(i) The ***TimingStatus*** of each stop used in the route. Stops may be deemed *principal points* or *time information points,* or both. The principal points must appear in a timetable, and so are mandatory for *TransXChange*, while other stop points are non-enforceable stops of the journey:

(ii) The ***Activity*** that takes place at each stop. For example, picking up or setting down passengers. This may need to be overridden on individual vehicle journeys.

TimingLink Properties			**JourneyPatternStopUsage**					
Link id	**Run Time [sec]**	**Dist-ance [m]**	**id**	**StopPoint Ref**	**Stop Name**	**Tim-ing Status**	**Wait Tim e**	**Activity**
JL_1 #1-#2	360	5.573	JL_1a	0600000001	Suborn, Bus Station	*PTP*	0	*PickUp*
			JL_1b	0600000002	Garden Village, Shops	*PTP*	0	*PickUpAndSetDown*
JL_2 #2-#3	300	4.512	JL_2a	0600000002	Garden Village, Shops	*PTP*	0	*PickUpAndSetDown*
			JL_2b	0600000003	Robridge, Plough	*PTP*	0	*PickUpAndSetDown*
JL_3 #3-#4	120	6.046	JL_3a	0600000003	Robridge, Plough	*PTP*	0	*PickUpAndSetDown*
			JL_3b	0600000004	Barford, Red Lion	*PTP*	60	*PickUpAndSetDown*
JL_4 #4-#5	180	2.520	JL_4a	0600000004	Barford, Red Lion	*PTP*	60	*PickUpAndSetDown*
			JL_4b	0600000005	Egham, Golden Lion	*TIP*	0	*PickUpAndSetDown*
JL_5 #5-#6	180	1.955	JL_5a	0600000005	Egham, Golden Lion	*TIP*	0	*PickUpAndSetDown*
			JL_5b	0600000006	Godhill, Church	*TIP*	0	*PickUpAndSetDown*
JL_6 #6-#7	420	2.963	JL_6a	0600000006	Godhill, Church	*TIP*	0	*PickUpAndSetDown*
			JL_6b	0600000007	Beall, Exchange	*PTP*	0	*PickUpAndSetDown*
JL_7 #7-#8	60	3.560	JL_7a	0600000007	Beall, Exchange	*PTP*	0	*PickUpAndSetDown*
			JL_7b	0600000008	Beall, Shell	*TIP*	0	*PickUpAndSetDown*
JL_7	60	2.005	JL_8a	0600000008	Beall, Shell	*TIP*	0	*PickUpAndSetDown*

#8-#9			JL_8b	0600000009	Beall, Bus Station	*PTP*	0	*SetDown*

Table 4-4 – Worked Example: Timing Links for Journey Pattern

4.8 Worked Example: Line

Line elements are used to model the labelling of lines for the public. A service may have a number of lines, each with a ***LineName***, and each vehicle journey can be assigned to a line. Normally, the same line is used to label vehicle journeys following the same pattern, but sometimes different journey variants with distinct patterns and link sequences may all be labelled under the same line name (though usually they will always have at least a few stops in common). Note that each ***VehicleJourney*** has a ***VehicleJourneyCode*** that is quite independent of the ***Line*** and ***LineName*** with which it may be associated.

In our example, there are three line names '1A'*, '1B'*, and *'1C'*, used to distinguish the different journeys that follow the three different journey patterns.

4.9 Worked Example: VehicleJourney

A ***VehicleJourney*** represents the traversal of a journey pattern at a particular time, and is composed of an ordered collection of ***VehicleJourneyTimingLink*** instances.

The timing links for ***VehicleJourney*** '40' of the worked example are shown in *Table 4-5*. Two stops are skipped: '*Robridge, Plough'* and '*Beall Bus Exchange'*.

TimingLink Properties			VehicleJourneyStopUsage					
id	**Run Time [sec]**	**JL Ref**	**id**	**StopPoint Usage**	**Stop Name**	**Stop Class**	**Wait Time**	**Activity**
VL_1 #1-#2	360	JL_1	VL_1a	0600000001	Suborn, Bus Station	BCT	0	pickUp
			VL_1b	0600000002	Garden Village, Shops	BCT	0	setDown
VL_2# 2-#3	300	JL_2	VL_2a	0600000002	Garden Village, Shops	BCT	0	pickUp
			VL_2b	0600000003	Robridge, Plough	BCT	0	**pass**
VL_3# 3-#4	120	JL_3	VL_3a	0600000003	Robridge, Plough	BCT	0	**pass**
			VL_3b	0600000004	Barford, Red Lion	BCT	60	setDown
VL_4# 4-#5	180	JL_4	VL_4a	0600000004	Barford, Red Lion	BCT	60	pickUp
			VL_4b	0600000005	Egham, Golden Lion	BCT	0	setDown
VL_5 #5-#6	180	JL_5	VL_5a	0600000005	Egham, Golden Lion	BCT	0	pickUp
			VL_5b	0600000006	Godhill, Church	BCT	0	setDown
VL_6 #6-#7	420	JL_6	VL_6a	0600000006	Godhill, Church	BCT	0	pickUp
			VL_6b	0600000007	Beall, Bus Exchange	BCT	0	**pass**
VL_7 #7-#8	60	JL_7	VL_7a	0600000007	Beall, Bus Exchange	BCT	0	**pass**
			VL_7b	0600000008	Beall, Shell	BCT	0	setDown
VL_7 #8-#9	60	JL_7	VL_8a	0600000008	Beall, Shell	BCT	0	pickUp
			VL_8b	0600000009	Beall, Bus Station	BCT	0	setDown

Table 4-5 – Worked Example: Timing Links for VehicleJourney 1A

4.10 Worked Example: Operational Times

The operational times of the example are modelled as follows:

- There is a ***Service*** / ***ValidityPeriod*** from '5th February 2001' until further notice.
- The ***OperatingProfile*** / ***RegularDayType*** / will show ***MondayToFriday*** operation.
- The ***OperatingProfile*** / ***BankHolidayOperation*** / ***DaysOfOperation*** / has a Value of ***MayDay*** to show that it runs on Mayday.
- The ***OperatingProfile*** / ***BankHolidayOperation*** / ***DaysOfNonOperation*** / has a Value of ***Christmas*** to show that it does not run on Christmas or Boxing Day.

5 EXAMPLES

5.1 Feature Examples

TransXChange is accompanied by a set of examples designed to illustrate the use of each of its features. For each example, a web page with links to the following is provided:

- Summary of features demonstrated by example.
- A route map.
- A matrix timetable representation.
- The XML encoding of the example.
- The *TransXChange Publisher* output of the encoded XML document.
- Explanatory notes describing the representation and implementation of specific features.

The examples can be found at http://www.transxchange.org.uk/examples.htm.
There is a different set of examples for each release of TransXChange.

Table 5-1 lists the *TransXChange* examples, with the features covered by each case.

Group	Name/ Topology	Features covered	Schema
Basic	**Linear** A single straight route. All vehicle journeys have the same timings	• Linear route. • Registration details. • RouteTrack maps • Tracks, including instructions and Mapping System References. • Frequency based journey times, specified as an interval. • Registration Schema. • Inbound & Outbound • Operator Details (including TXC v2.4 attributes) • AssistanceService (including TXC 2.5 attributes) • VehicleAccessEquipment (including TXC 2.5 attributes)	Registration
	Express A linear route with express journey patterns	• Express service. • Reuse of vehicle journey timing links in multiple journeys. • Overriding of Journey Pattern Run times, • Holiday day type exclusions. • Local stop point definitions for an off-street bus station: BCQ and BCS stop types. • Local stop area definitions. • Variable bay allocation. • Supporting document. • Journey Footnotes, Marketing Name	General
	Cancellation (TXC 2.4)	• Basic Cancellation details • Registration Operation & service only.	Registration
Complex	**Interchange** Two patterns run by two different operators presented as the same Service. All vehicle journeys have the same timings.	• An Interchange. • Linear route, with different stop visiting pattern at one end. • Express stops. • Frequency based journey times, specified as an interval. • Combining operating days from service, journey pattern and vehicle journey level. • Serviced organisation & school dates. • More than one operator. • Timetable notes. • Inward and outward timetables for the same service, using a single route. • Registration Schema.	Registration

	Circular A circular route.	• Circular route. • Reuse of route sections. • Dead runs (positioning links). • Partial traversal of a journey pattern. • Operator Garage. • AVL data - Vehicle type ticket machine, duty crew. • WGS84. • Block (Running Board). • General Schema.	
	Cloverleaf A cloverleaf route shape with three petals	• Multiple routes composed of common route sections. • Multiple journey patterns composed of common. Journey pattern sections. • Dynamic destination display. • General Schema.	
	Lollipop A lollipop" shaped route, with two parallel branches.	• Circular and parallel sections. • Reuse of journey pattern sections. • Stop Sequence Numbers. • Layover Point. • Connecting services. • General Schema.	
	Eye An 'eye' shaped route, with two alternative branches.	• Multiple routes composed of common route sections. • Multiple journey patterns composed of common. Journey pattern sections. • Stop Sequence Numbers. • Local stop point definitions. • Bilingual stop names & schedule (Cymraeg). • Block (Running Board). • Registration Schema.	Registration
	Flexible Use of flexible zones	• Flexible zones. • Flexible time bands. • Registration Schema.	Registration
	Footnotes Use of footnotes	• Flexible zones. • Frequent Journeys • Large number of services (144) • Page overflow. • Garage detail • Short Notice Registration • Operational Data	Registration
	Hail & Ride Use of hail and ride stops	• Hail and ride sections. • Local stop point definitions. • Full lollipop topology. • Frequency based journey times, specified as minutes past the hour (but not a Frequent Service). • Short notice registration details. • Scottish Bank Holidays. • Workflow attributes • AccessibilityBooking (TXC v2.5)	Registration
	Large Route Very large timetable	• More stops than fit on a page. • More journeys than fit on a page.	Registration
	Merge Frequent Journeys	• Combing of individual journeys into a single column. • Page overflow. • Non PTP points. • Use of default operating Profile.	Registration
	Grouping (+TXC v2.4) Journey Groupings	• Use of Journey Grouping (+TXC v2.4) • Built-in Journey Grouping Labels. (+TXC v2.4) • Custom Journey Groupings. (+TXC v2.4) • Calendars (+TXC v2.4)	Registration

		• Data Rights (+TXC v2.4) of an operator. • Data Rights (+TXC v2.4) of a contributor. • Validity conditions.	
	Operators Operator data only	• Exchange of just Operators	General
	Routes Route Data only	• Exchange of just Routes	General
Delta	**Delta** (+TXC v2.4) Changes to timetable	• Exchange of elected changes	General Delta

Table 5-1 – TransXChange Examples

5.2 Real Examples

A small number of ‘real’ examples, contributed by *TransXChange* users are also included on the web site. These provide larger examples using realistic data on an as is basis.:

Name	Features covered	TXC	Courtesy
U10_SOX_PH_U5_20090701	• Cancellation. • Multiple lines within the service; timetables marked with comments defining operating periods (university term/vacation). • Co-ordinates in WGS84. Daily variations.	2.1	Stagecoach
3_VI_PF_3_20071028	• Lollipop " route; • Frequent interval covering part of length of route so described in an attachment. • Co-ordinates in WGS84.	2.1	Stagecoach
7_SSS_PB_78_20080727	• Frequent intervals assigned to trips and also timetable described as frequency, • Holiday timetables, • Circular services. • Co-ordinates in WGS84.	2.1	Stagecoach
86_STA_PD_R86_20070903	• Co-ordinates in Eastings/ Northings.	2.1	Stagecoach
S5_SOX_PH_S5_20090125.zip	• Complex service with route variations. • Monday-Friday timetable split into Monday-Wednesday/Thursday/Friday reflecting operational requirements. • Bank holiday operating profiles.	2.1	Stagecoach
X5_ST_PF_785_20080426	• Limited stop service; attachment showing alternative routes (not affecting stopping places). • Co-ordinates in WGS84.	2.1	Stagecoach

Figure 5-1 – Table List of TransXChange real examples

6 TRANSXCHANGE SCHEMA

TransXChange describes bus schedules as a model of XML elements, contained within a ***TransXChange*** root element. In this section we describe the different types of schema elements in detail.

6.1 TransXChange Schema Overview

In a *TransXChange* document, data is organised around two main element types; ***Service*** (which may contain either or both ***StandardService*** or ***FlexibleService*** elements) and ***VehicleJourney***, which together combine the instances of other elements into descriptions of bus schedules. ***Service*** instances are grouped under the ***TransXChange*** root element within a ***Services*** container, and ***VehicleJourney*** instances in a ***VehicleJourneys*** container. Other high-level elements such as ***Operator, Registration, RouteSection*** and ***JourneyPatternSection*** are also declared globally within containers under the ***TransXChange*** root element so that they may be reused in many different services (or even outside the context of a service, for general exchange).

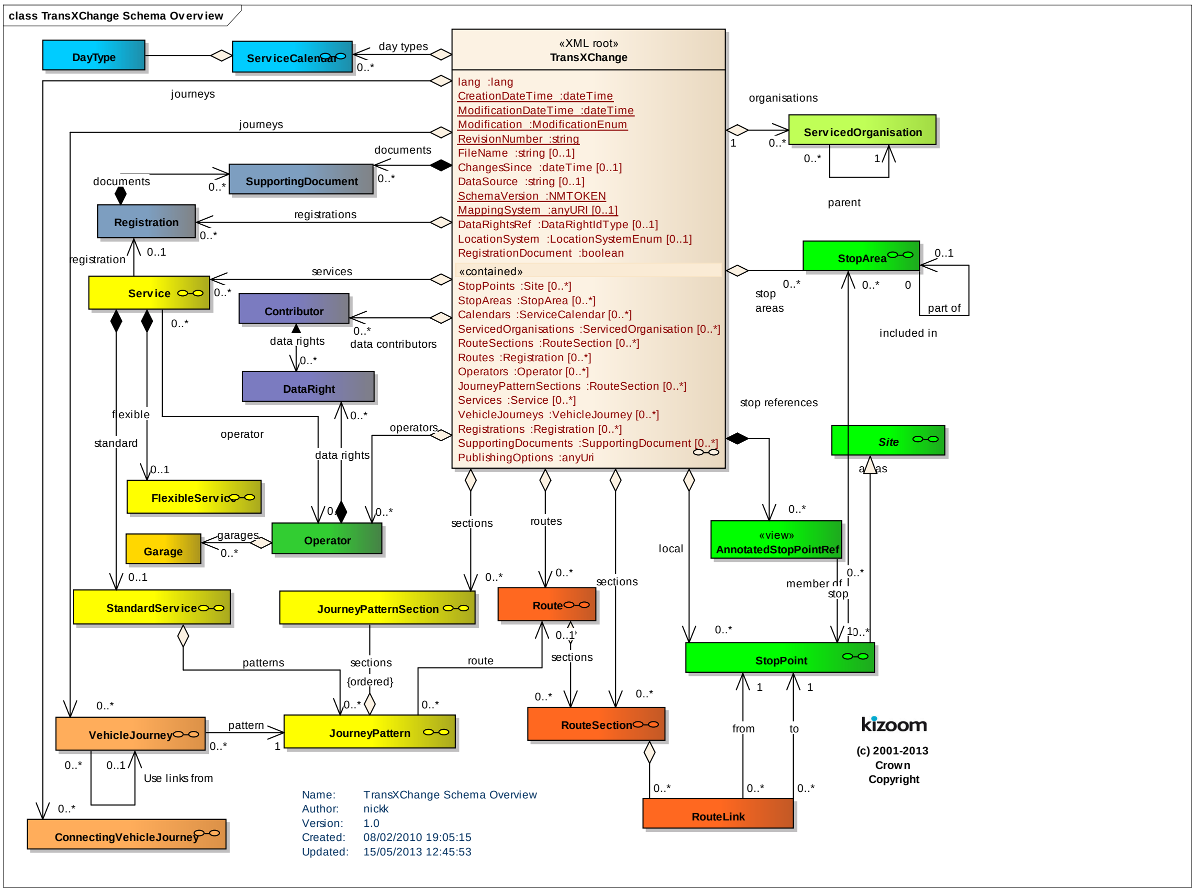


Figure 6-1 – UML Diagram of TransXChange Schema: Core elements

The ***TransXChange*** element thus contains a number of different child and descendant elements (*Figure 6-5*) which can be characterised as falling into four groups:

- *Topographical* elements: ***StopPoint***, ***StopArea***, ***NptgLocality***, ***ServicedOrganisation.***
- *Route* and Network *topology* elements: ***Route***, ***RouteSection, RouteLink***.
- *Service Supply* elements: ***Service***, (***StandardService***, ***FlexibleService, Line, JourneyPattern), JourneyPatternSection, VehicleJourney.***
- *Registration Elements:* ***Operator, Registration, (ShortNoticeRegistration).***
- *Ancillary* elements: ***SupportingDocument***. This can be specified at either the document or the registration level

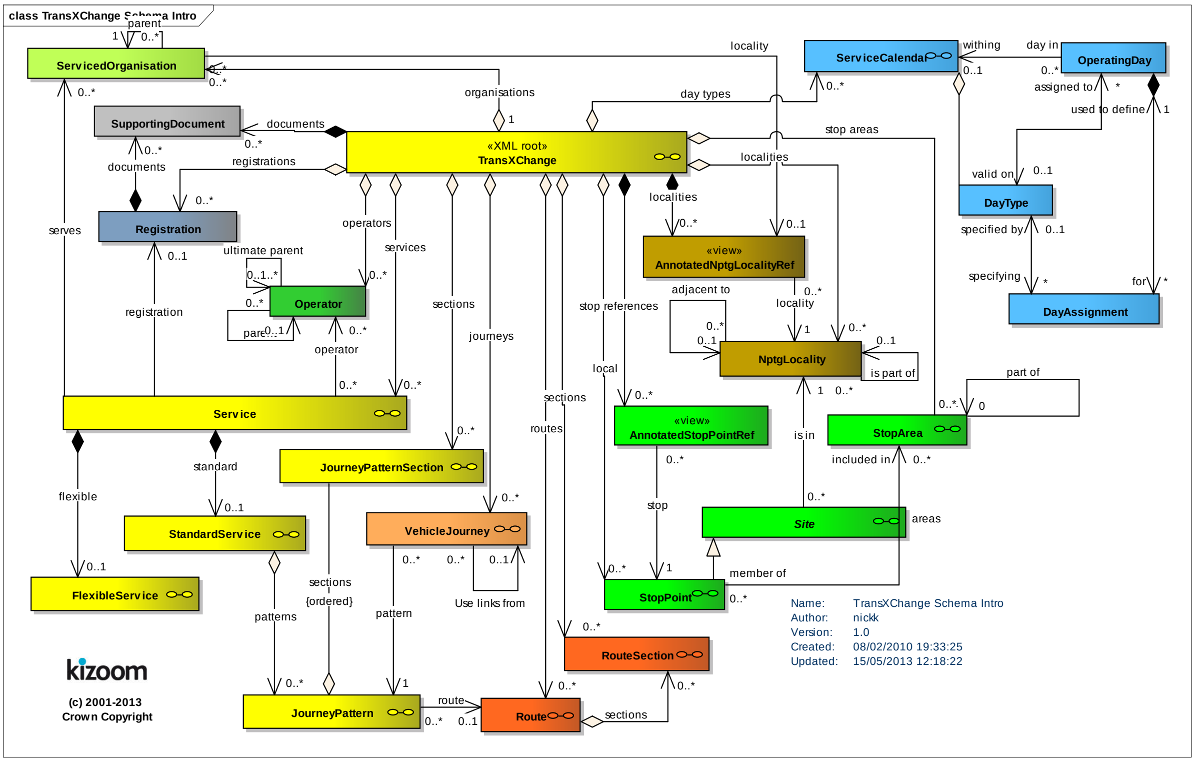


Figure 6-2 – UML Diagram of TransXChange Schema Overview of elements

6.2 TransXChange Root Element

Every *TransXChange* document has a single instance of the ***TransXChange*** root element, which contains all the other elements.

- The *TransXChange attributes* specify document level attributes. See below.
- *TransXChange* Element Group: There are different Groups for the Registration & general schemas, reflecting slight different constraints on the registration & general elements.
- The ***PublishingOptions*** allows presentation options to be included. These can be described by a different embedded schema
- The *Constraints* specify referential integrity rules. See Later.

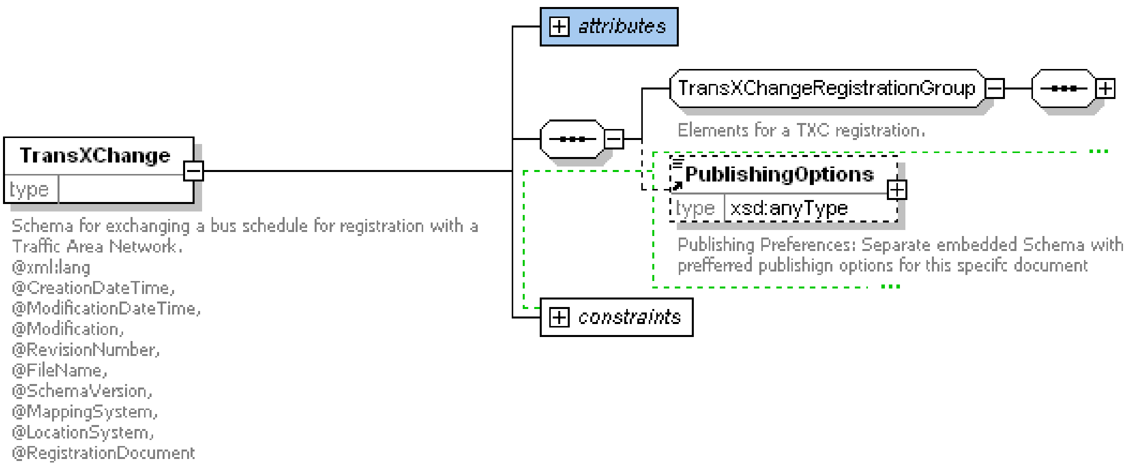


Table 6-1 – TransXChange Root element

6.2.1 TransXChange Element Attributes

The ***TransXChange*** element has the following attributes:

- Document Modification Attributes group
  - ***xml:lang:*** Default language of document. ISO language identifier. Default is English.
  - ***CreationDateTime***: Timestamp of document creation date and time.
  - ***ModificationDateTime:*** Timestamp of document last modification date and time.
  - ***Modification:*** Nature of update ***'****New', 'Revise', 'Delete'****.***
  - ***RevisionNumber:*** Monotonically increasing version number.
  - ***FileName:*** Name of file containing the document.
  - ***ChangesSince:*** If file contains only delta changes since a given date (TXC v2.4)
- Document Metadata Attributes group
  - ***SchemaVersion:*** *TransXChange* schema version identifier used for the document content model. Fixed: must be the schema version, e.g. *'2.0'*.
  - ***MappingSystem:*** Data system to use for mapping references ('*OS', 'Navtech'*, etc) within the document.
  - ***DataSource: Name of provider of the data***
- Fixed Attributes
  - ***LocationSystem:*** Data system to use for location coordinate references within the document: '*WGS84'* or '*Grid'*. Must be '*Grid'* for registration documents.
  - ***RegistrationDocument:*** Whether the document should be published as a registration, i.e. satisfy the additional semantic integrity constraints. Boolean.

6.2.2 TransXChange Root Attributes

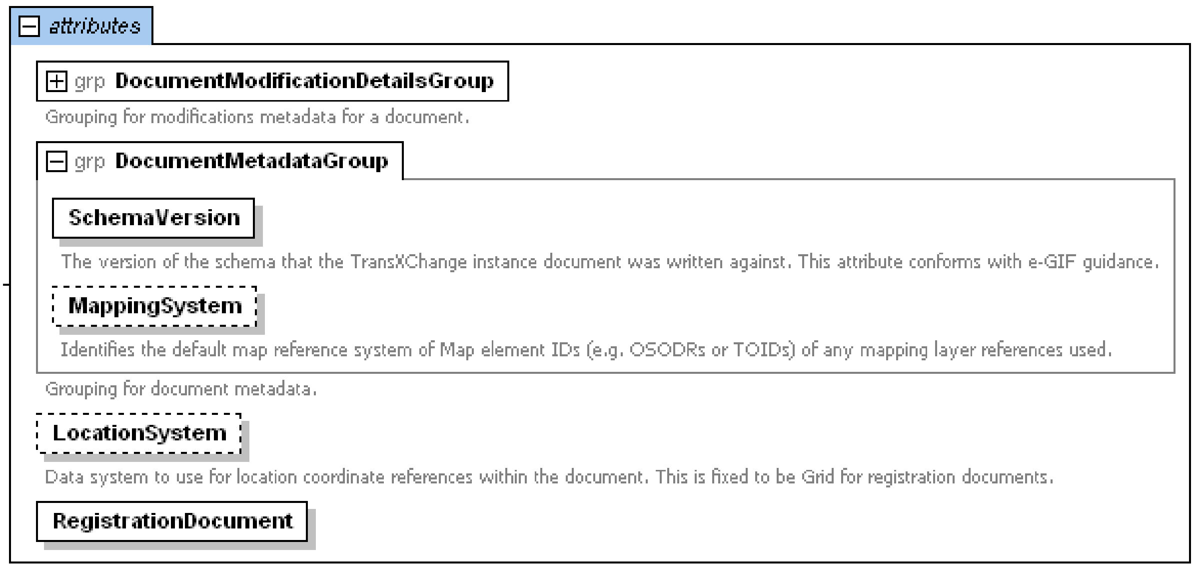


Figure 6-3 – Attributes for TransXChange Schema

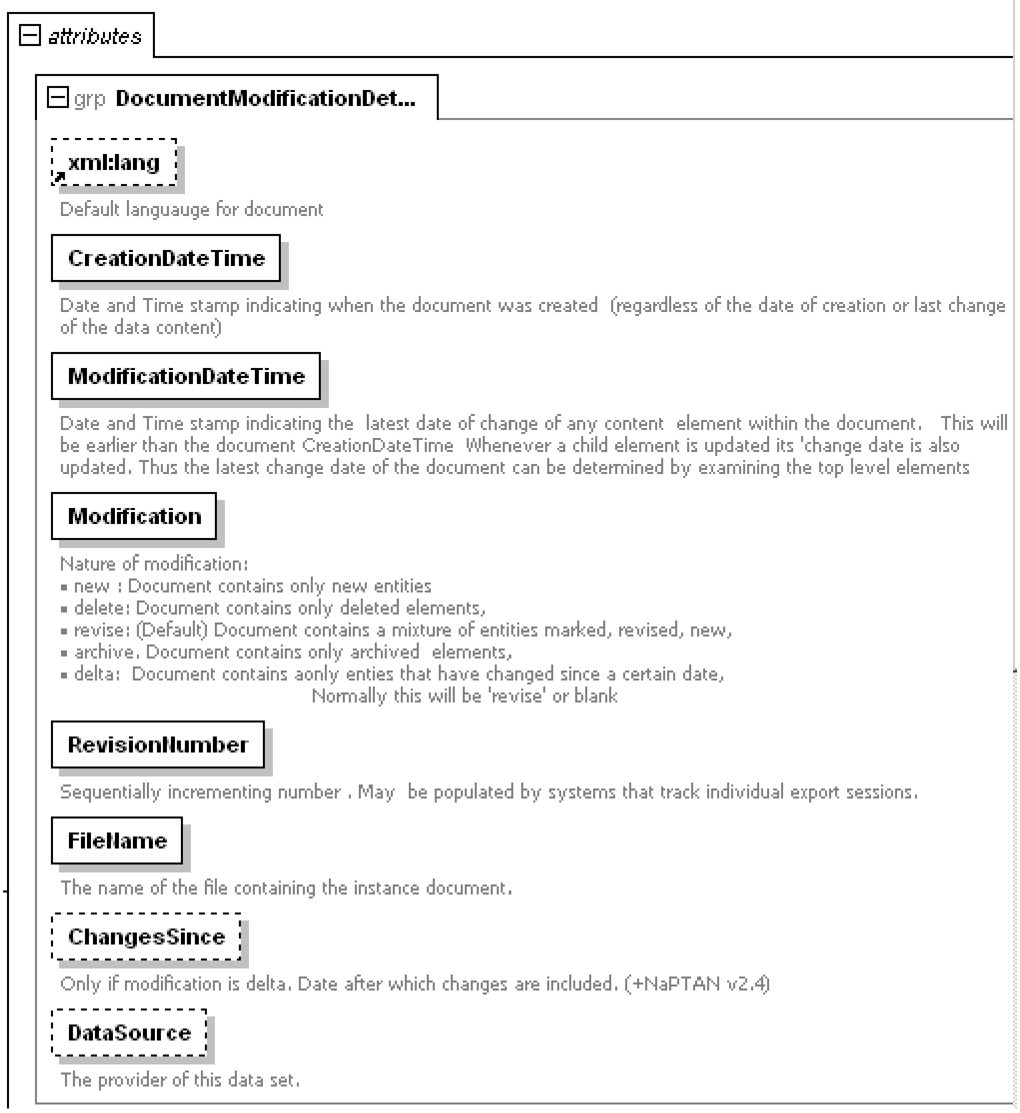


Figure 6-4 – TransXChange Change management attributes

6.2.3 TransXChange Child Elements

The ***TransXChange*** element contains a group element comprises the following child elements:

- ***ServicedOrganisations***: A collection of ***ServicedOrganisation*** elements. See later.
- ***NptgLocalities***: A collection of references to NPTG *localities* used in local stop definitions in the schedule. See later.
- ***StopPoints***: A collection of the *NaPTAN* stop points used in the schedule. See later.
- ***StopAreas***: A collection of reusable ***StopArea*** instances declared locally to group any stop points declared locally. See later.
- ***RouteSections***: A collection of reusable ***RouteSection*** elements for defining routes. See later.
- ***Routes***: A collection of reusable ***Route*** elements for use in journey patterns. See later.
- ***JourneyPatternSections***: A collection of reusable ***JourneyPatternSection*** elements for defining journey patterns. See later.
- ***Operators***: A collection of ***Operator*** elements. See later.
- ***Services***: A collection of ***Service*** elements. See later.
- ***VehicleJourneys***: A collection of ***VehicleJourney*** elements. See later.
- ***Registrations***: A collection of ***Registration*** elements, each referencing a ***Service*** element. See later.
    - In the *TransXChange Registration Schema*, there must be one ***Registration.***
    - In the *TransXChange General Schema* documents, there may be zero, one or many ***Registration*** instances.
- ***SupportingDocuments***: A collection of reusable ***SupportingDocument*** elements. See later.

Figure 6-5 – Top Level Elements of TransXChange

6.3 Stop & Topographical Elements – StopPoints and Zones

6.3.1 NptgLocalities Element

The use of stops in *TransXChange i*s based on *NaPTAN.* See the ***StopPoints*** element which allows stop usages to be declared. All stops are assigned to an *NPTG* Locality by means of a reference to a NPTG Locality identifier. When publishing the stop with a tool such as the *TransXChange Publisher*, the bus stop common names may be qualified with a locality name, for example "*Barset, High Street",* rather than just *"High Street".* It is therefore desirable that a *TransXChange* document contain the NPTG locality names so that a document can be published without recourse to the NPTG database. For stops that are externally referenced (using an ***AnnotatedStopPointRef*** instance), the ***NptgLocality / LocalityName*** can be included as an annotation on the stop point reference. However for new stops that are defined locally using a ***StopPoint*** element, the locality names need to be supplied with a separate ***AnnotatedNptgLocalityRef,*** as they are not part of a new ***StopPoint*** declaration.

- ***NptgLocalities***: A collection of ***AnnotatedNptgLocalityRef*** instances. See below.

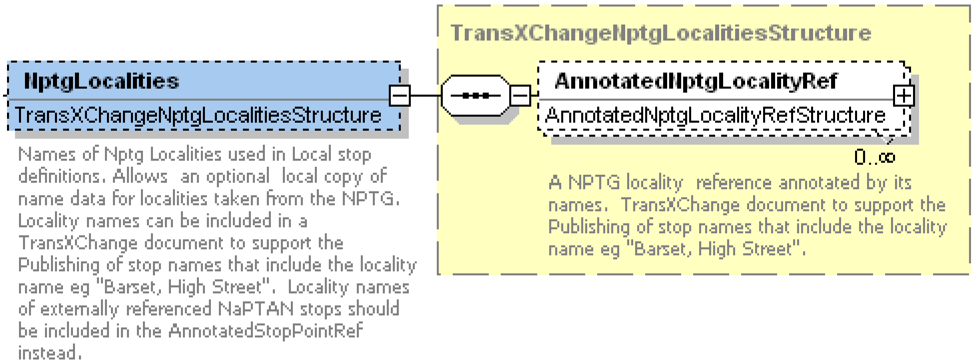


Figure 6-6 – NptgLocalities Element

6.3.2 AnnotatedNptgLocalityRef Element

Each ***AnnotatedNptgLocalityRef*** instance provides a local copy of *NPTG* Locality name information that can be used without recourse to the full *NPTG* database.

- ***NptgLocalityRef:*** Unique *NPTG* Locality identifier, i.e. ***NptgLocalityCode*** of ***NptgLocality***
- ***LocalityName:*** Text name of ***NptgLocality;*** this name can be repeated locally so that the schedule may be annotated by tools such as the *TransXChange Publisher* without necessarily accessing the full *NPTG* database.
- ***LocalityQualifier:*** Any Qualifier of text name of locality, for example "Kent" to distinguish '*Ashford (Kent)'* from '*Ashford (Middlesex)*.

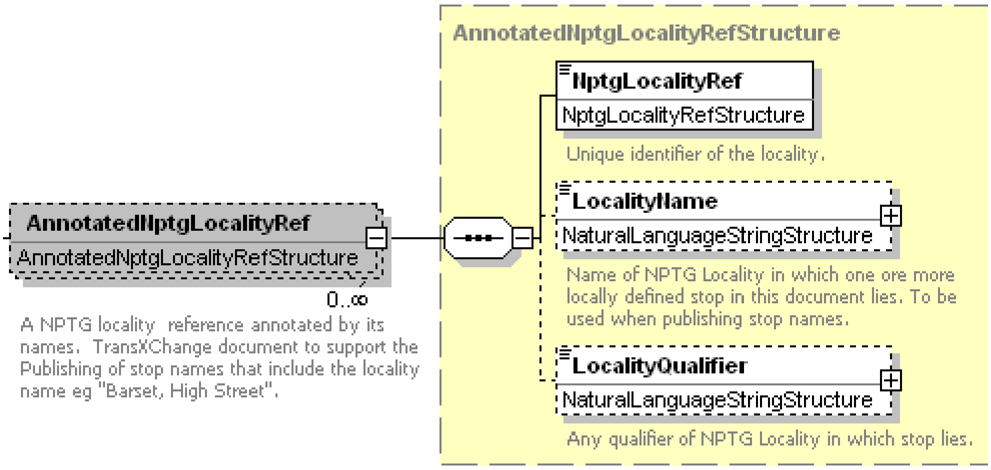


Figure 6-7 – AnnotatedNptgLocalityRef Element

6.3.3 StopPoints Element

The use of stops in *TransXChange is* based on *NaPTAN.* The ***StopPoints*** element (*Figure 6-8)* contains reusable declarations of the stops used by the routes and journey patterns of the schedule. All ***StopPointRef*** instances elsewhere in a document are resolved against the contents of the ***StopPoints*** element.

- Existing *NaPTAN* ***StopPoint*** instances can be referred to simply by using an ***AnnotatedStopPointRef*** to reference a *NaPTAN* system stop identifier – the ***AtcoCode*** of the stop. For further details refer to the *NaPTAN Schema Guide.*
- Stops may also be declared within a *TransXChange* XML document, by means of a local ***StopPoint*** declaration within the ***StopPoints*** container element. This can be used to provide full descriptive details of a new stop not yet in the NaPTAN database, or to correct details about an existing stop.

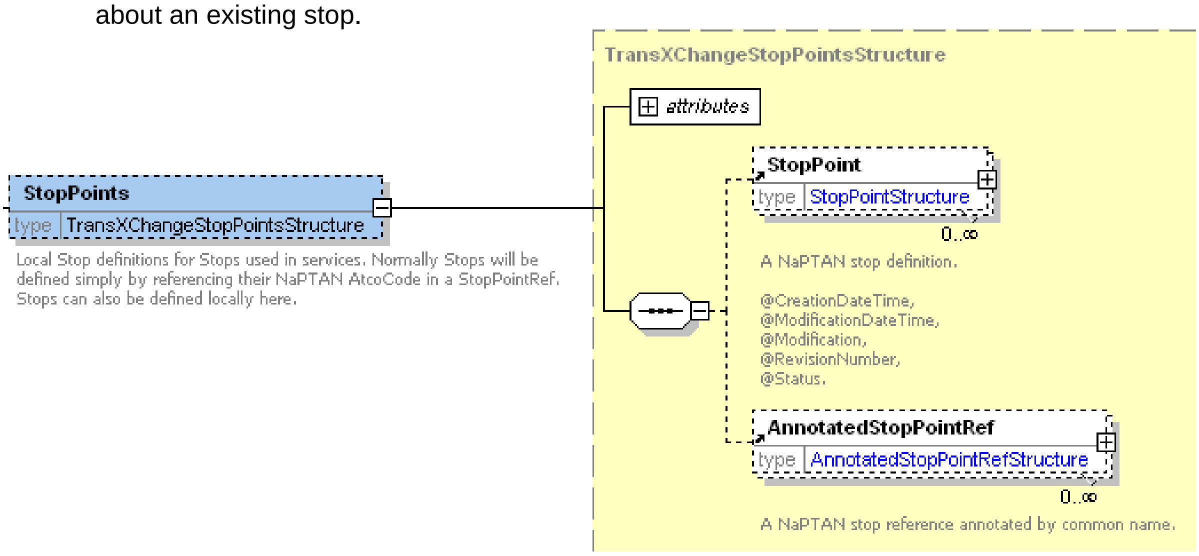


Figure 6-8 – StopPoints Element

6.3.4 AnnotatedStopPointRef Element

The ***AnnotatedStopPoint*** element (*Error! Reference source not found.*) references an existing *NaPTAN* stop and comprises the following elements:

- ***StopPointRef:*** Unique *NaPTAN* identifier, i.e. ***AtcoCode*** of ***StopPoint***.
- ***CommonName:*** Common text name of ***StopPoint;*** this name is repeated locally so that the schedule may be interpreted by tools such as the *TransXChange Publisher* without necessarily accessing the full *NaPTAN* database.

- ***Indicator:*** Further structured text descriptor element of ***StopPoint;*** that is used to distinguish similar stops, for example bus station bays.
- ***LocalityName:*** Text name of ***NptgLocality;*** this name can be repeated locally so that the schedule may be annotated by tools such as the *TransXChange Publisher* without necessarily accessing the full *NPTG* database.
- ***LocalityQualifier:*** Any Qualifier of text name of locality, for example "Kent" to distinguish '*Ashford (Kent)'* from '*Ashford (Middlesex)*.

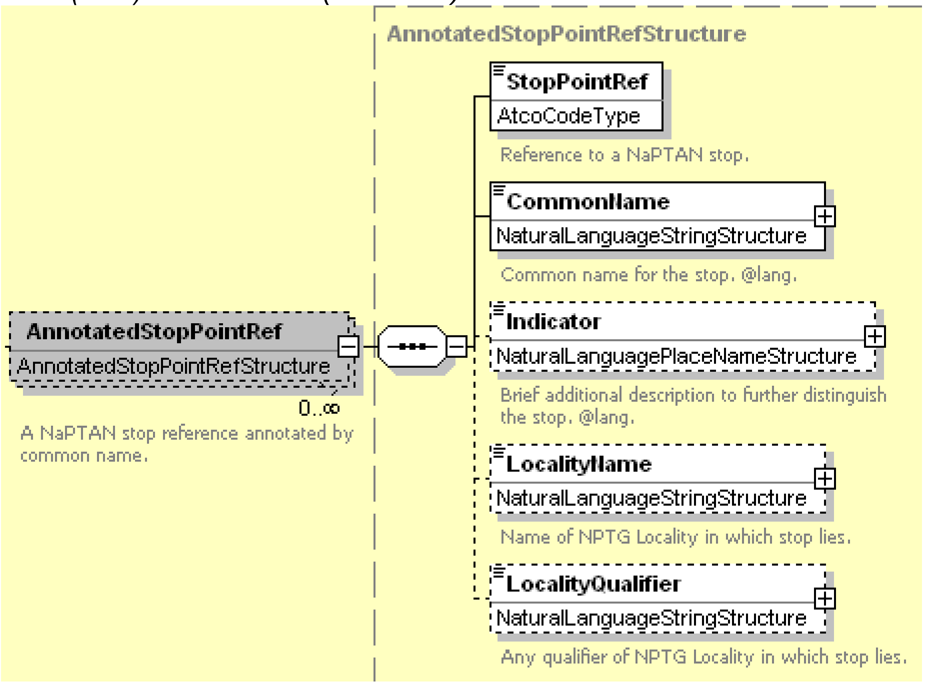


Figure 6-9 – Annotated StopPointRef Element

6.3.5 StopPoint Element (Stop)

The ***StopPoint*** element declares locally defined stops. A local ***StopPoint*** declaration uses *NaPTAN* elements, and must include a *NaPTAN* identifier for the stop. Local declarations are for expediency in cases when the *NaPTAN* definition for a new stop (or a change to an existing stop) has not yet been promulgated to the *NaPTAN* database. Even then, the *NaPTAN* identifier for such stops must be allocated by the relevant local transport authority. The other details of the stop may change subsequently in the course of registering it with the Authority.

Refer to the *NaPTAN* 2.0 *Schema Guide* for a definition of the ***StopPoint*** Element and its subelements.

6.3.6 StopArea Element

A ***StopArea*** is used to group stops: locally declared ***StopPoint*** instances can be assigned to one or more stop areas.

- *NaPTAN* stops that exist in the *NaPTAN* database may already have a ***StopArea*** element (previously called a ***StopGroup***) associated with them.
- Local definitions of individual ***StopArea*** elements may also be declared within the ***StopAreas*** element of the ***TransXChange*** root element. Each ***StopArea*** must have a ***StopAreaCode***. Local stop area declarations are for expediency in cases when the *NaPTAN* definition for a new stop area has not yet been promulgated to the *NaPTAN* database.
- Locally declared ***StopPoint*** elements may reference one or more ***StopArea*** instances.
- When importing schedules, an application will attempt to find the ***StopArea*** details in the *NaPTAN* database using the ***StopAreaCode***. Only if no ***StopArea*** is found for the code will the locally supplied definition be used.

A *NaPTAN* ***StopArea*** is identified by an ***AreaCode,*** a unique *NaPTAN* identifier of the stop area.

Refer to the *NaPTAN* 2.0 *Schema guide* for a definition of the ***StopArea*** Element and its subelements.

6.4 Network Elements – Routes and Tracks

6.4.1 Route Element

A ***Route*** (*Figure 6-10*) describes the physical traversal of a bus along a route, described as an ordered collection of ***RouteLink*** elements, grouped into ***RouteSection*** elements. It is identified by a unique ***id*** attribute, and has the following properties:

- ***PrivateCode***: an optional cross reference to an external system identifier for the route.
- ***Description:*** A textual description of the route.
- ***RouteSectionRef***: An ordered collection of one or more references to ***RouteSection*** elements that contain the route links making up the route.
- ***ReversingManouevre:*** Used to describe any reversing manoeuvres needed.

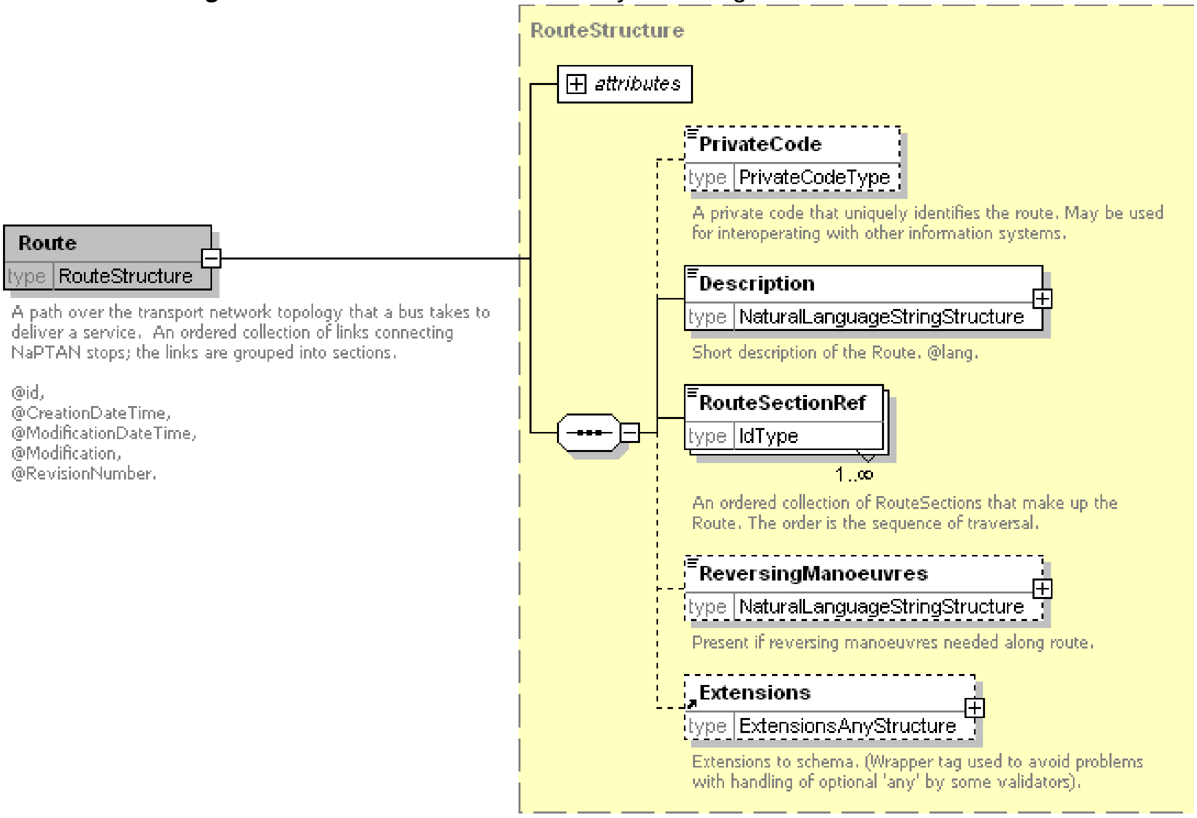


Figure 6-10 – Route Element

6.4.2 RouteSection Element

A ***RouteSection*** (*Figure 6-11*) describes the course of a section of a route between several *NaPTAN* stops, and comprises an ordered collection of ***RouteLink*** elements, each describing a stop-to-stop path. A ***RouteSection*** can be used in multiple routes. It is identified by a unique ***id*** attribute.

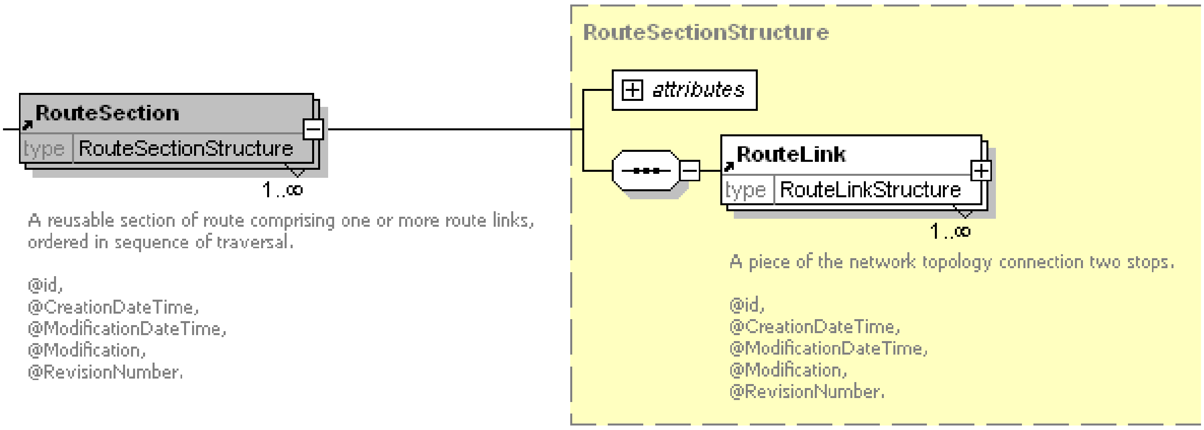


Figure 6-11 – RouteSection Element

6.4.3 RouteLink Element

A ***RouteLink*** (*Figure 6-12*) describes the course of a route between two *NaPTAN* stops. It is identified by a unique ***id*** attribute, and comprises:

- ***RouteRef:*** Optional reference to parent ***Route***. Normally not stated as given by containing context, but may be specified when using the ***RouteLink*** as a stand-alone artefact. If already given by context, this value is ignored**.** (+TXC v2.4).
- ***From***: The ***StopPointRef*** to the stop at which the link starts.
- ***To***: The ***StopPointRef*** to the stop at which the link ends.
- ***Distance***: The length of the path along the route in meters.
- ***Direction***: Direction of the ***Route*** running over the ***RouteLink***. See *Table 6-2*.

Value	Description
inbound	Inbound Direction.
outbound	Outbound Direction.
clockwise	Clockwise Direction.
antiClockwise	Anti-Clockwise Direction.

Table 6-2 – Allowed Values for Link / Direction

- ***Track***: A description of the path of the link as one or more ***Track*** elements.

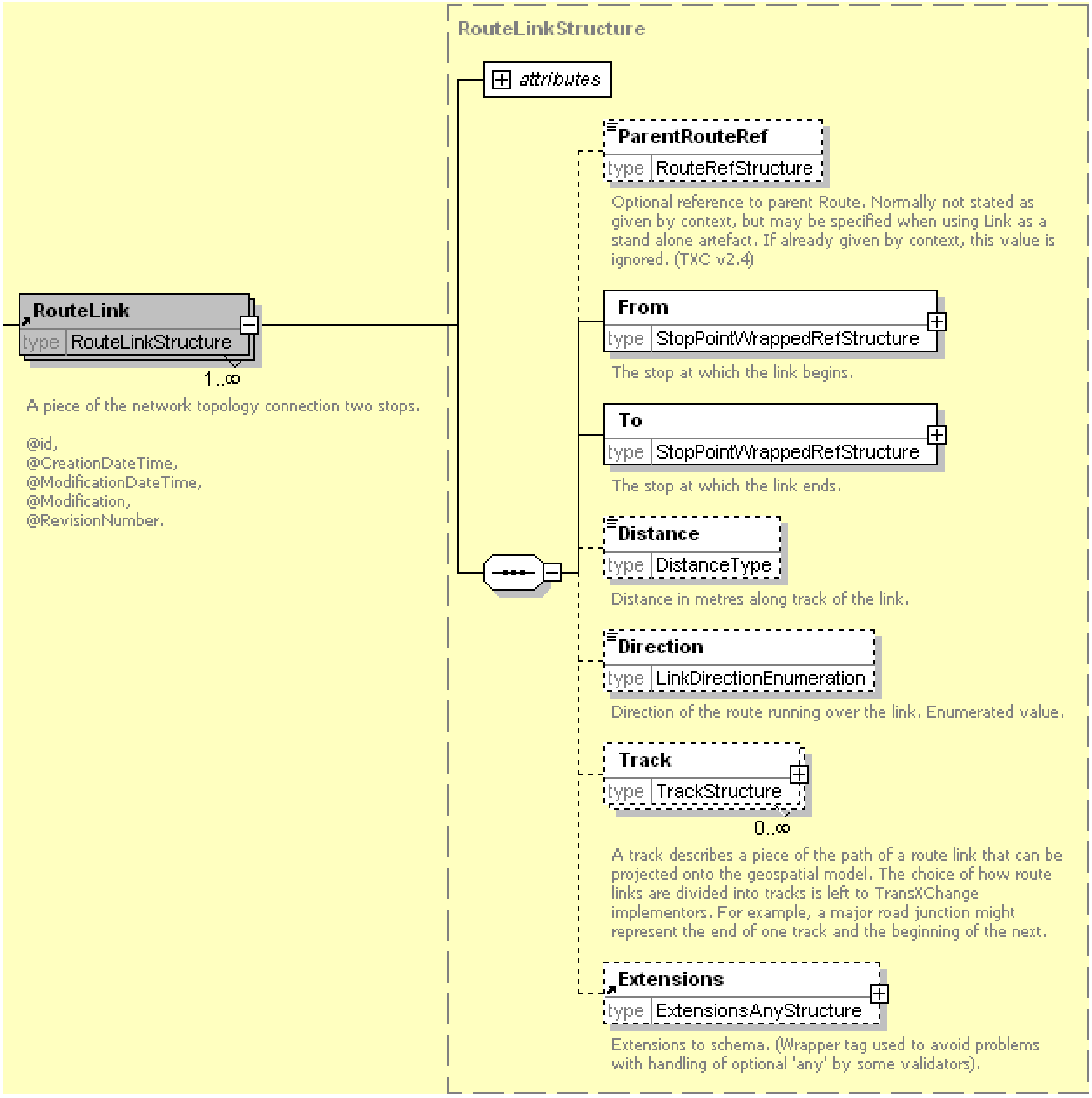


Figure 6-12 – RouteLink Element

6.4.4 Route / Track Element

A ***Track*** element (*Figure 6-13*) describes the path of a route link between *NaPTAN* stops, and optionally, intermediate junction points. It comprises:

- ***RouteLinkRef:*** Optional reference to parent ***RouteLink***. Normally not stated as given by containing context, but may be specified when using the ***RouteLink*** as a stand-alone artefact. If already given by context, this value is ignored**.** (+TXC v2.4).
- ***Mapping***: A description of the path of the route as a series of geospatial points.
- ***MapSystemReference***: An optional reference to an Ordnance Service TOID or other map feature identifier, using the mapping data system specified by the **MappingSystem** attribute.
- ***Instructions***: Optional detailed step-by-step text instructions for navigating the track.

It is up to the implementer to choose the granularity of tracks – a give route might be represented by none, one, several or many tracks. Typically a track will be used for each distinct road or mapping layer feature that the implementer wishes to associate with part of the route.

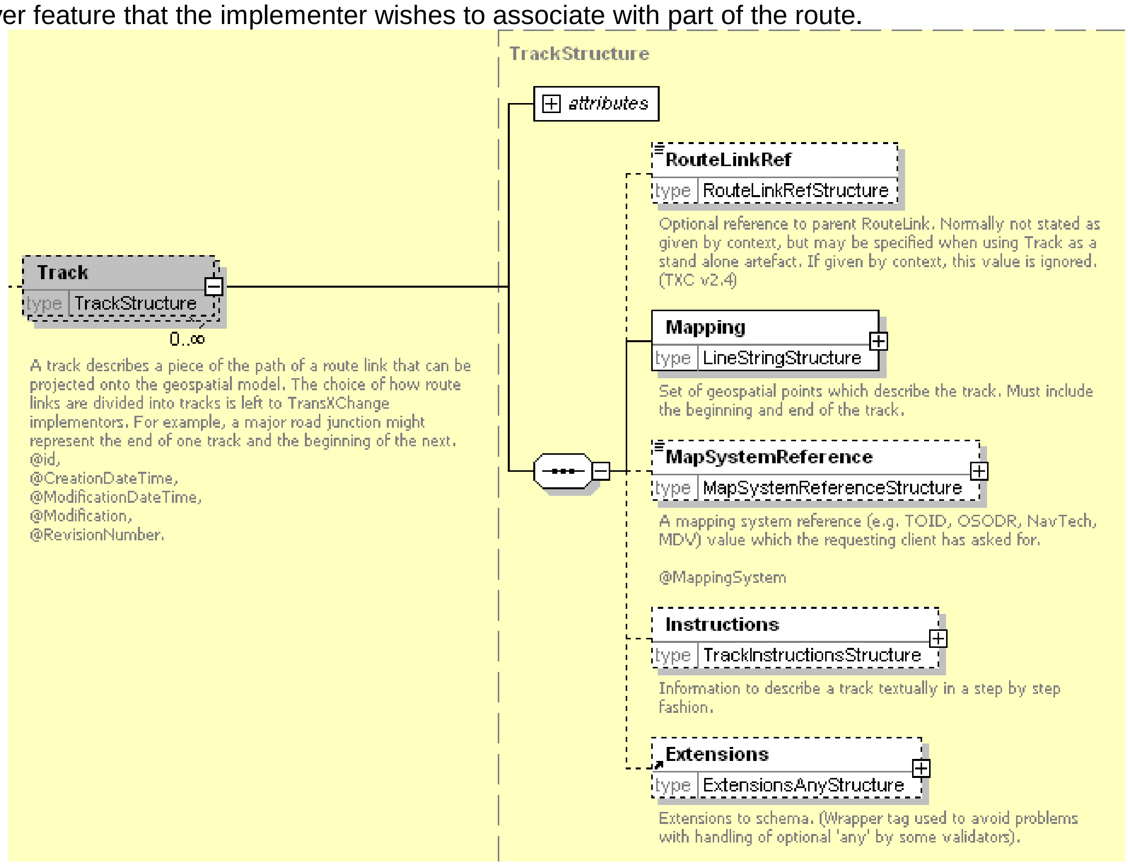


Figure 6-13 – Track Element

6.4.5 Track Subelements

6.4.5.1 Track / Mapping Element

A ***Mapping*** element (*Figure 6-14*) describes the spatial path of a route link between *NaPTAN* stops that can be plotted on a map, as a series of at least two geospatial points: These points are independent of the stop point coordinates (though end points may reference the same coordinate) I.e. to plot a route the last and first point of each successive mapping will be connected.

- ***Location***: A point in either WGS84 or grid coordinates. See Common Schema Elements later.

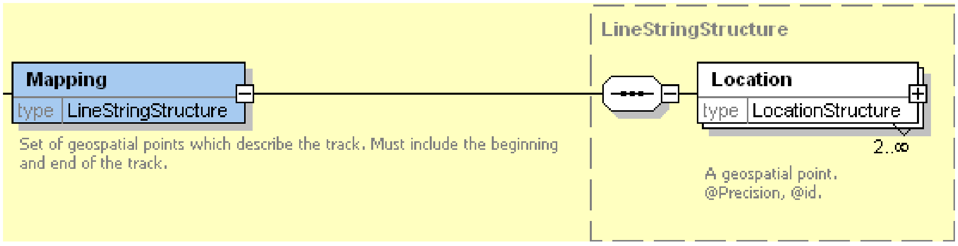


Figure 6-14 – Mapping Element

6.4.5.2 Track / Instructions Element

The ***Instructions*** element (*Figure 6-15*) provides an additional description of the path of a step of a route between *NaPTAN* stops as text instructions, and an ordered collection of structured elements:

- ***Summary***: A free text description of the path of the route.
- ***Feature***: A structured description of one or more steps of the journey.

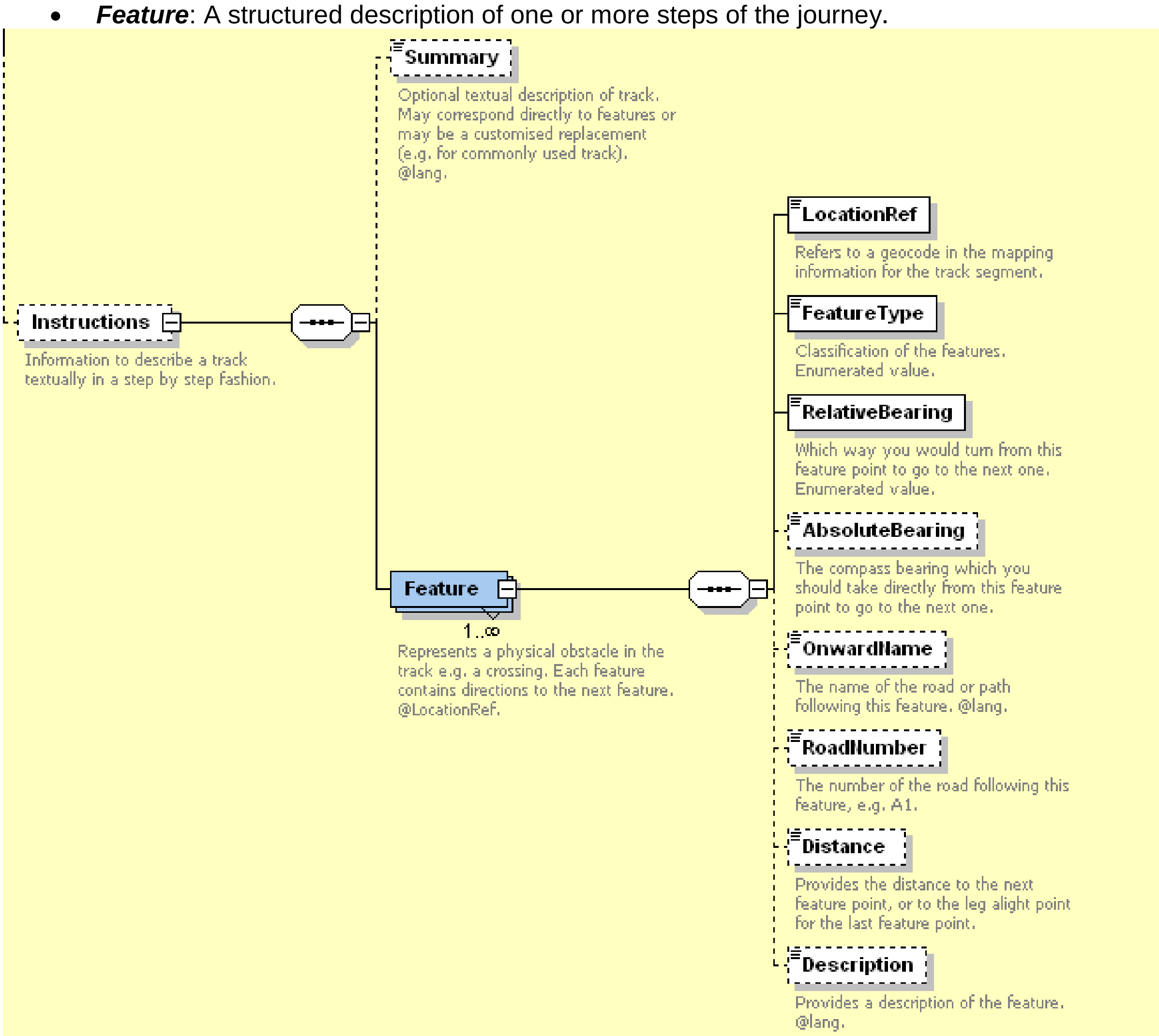


Figure 6-15 – Instructions Element

6.4.5.3 Track / Instructions / Feature Element

The ***Feature*** element (*Figure 6-15*) describes a step of a route between *NaPTAN* stops:

- ***LocationRef***: Reference to a ***Location*** in the ***Track's Mapping*** instance that locates the feature on a map.
- ***FeatureType***: Describes the type of feature encountered see Table 6-3.

Value	Description

legOrigin	The start point of the leg.
legDestination	The end point of the leg.
bend	A bend in the track that merits attention (without a junction).
crossing	Cross over the road.
bridge	Traversing over a bridge.
junction	Either a point at which another road is taken, or a side road that is passed along the way.
miniRoundabout	Going round a small roundabout.
roadChange	Denotes a change of road name when there is no junction.
roundabout	Going round a small roundabout.
subway	Going through a subway.
trafficLights	Going through traffic lights.
landmark	A named landmark that can be seen from the track. The name should be provided in the Feature Description.

Table 6-3 – Allowed Values for FeatureType

- ***RelativeBearing:*** Which way you would turn from this feature to go to the next one. See *Table 6-4.*

Value	Description
left	Left
right	Right
straightAhead	Straight ahead
uTurn	U-turn

Table 6-4 – Allowed Values for RelativeBearing

- ***AbsoluteBearing:*** The compass bearing which you should take directly from this feature point to go to the next one.
- ***OnwardName***: The name of the road or path following this feature
- ***RoadNumber***: The number of the road following this feature, e.g. ‘A1’.
- ***Distance***: The distance to the next feature point, or to the leg alight point for the last feature point.
- ***Description***: A text description of the individual feature.

6.5 Registration Elements: Operator, Registration, ShortNoticeRegistration

6.5.1 Operators Element

The ***Operators*** element (*Figure 6-16*) contains instances of the two different kinds of operator element that may be referenced by a ***Service:***

- ***Operator:*** An operator definition allowing partial definition of an operator. See below.
- ***LicensedOperator:*** A full definition of an operator as is required for a registration.

In addition it can also be used to declare other types of organisation:

- ***Contributor:*** A holder of data rights other than an operator***. See below.***

The ***Operator*** and ***LicensedOperator*** elements differ only as which of their child elements are required or optional.

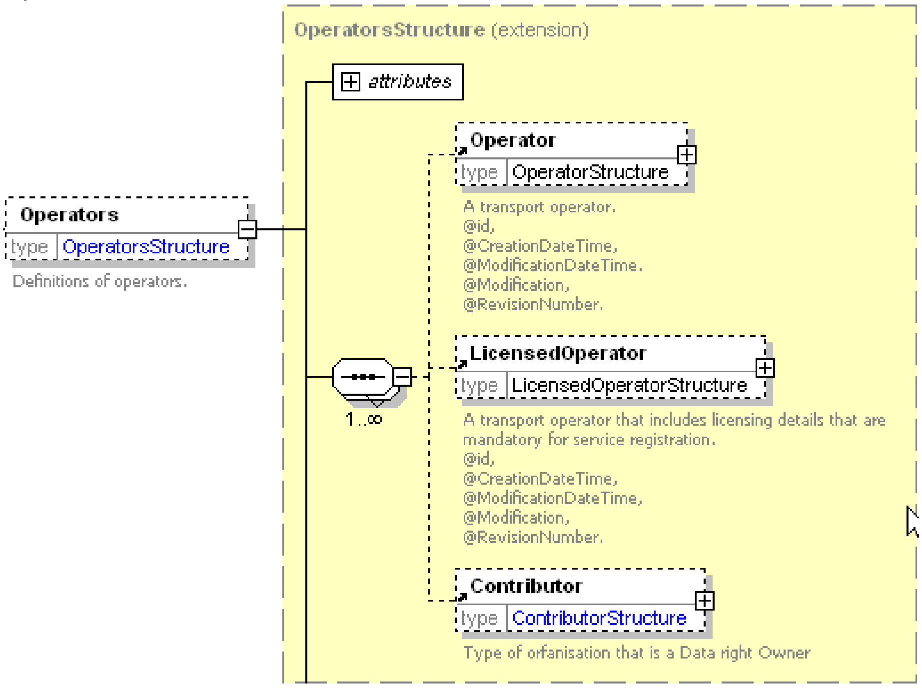


Figure 6-16 – Operators Element

Figure 6-17 – Operators Element

6.5.2 Operator Element

The ***Operator*** element (*Figure 6-18*) describes the ***Operator*** of a service. Every operator has an **id** attribute. References to operators within the document are made through the **id** (rather than the ***OperatorCode*** or the ***NationalOperatorCode),*** in order to guarantee a unique reference. ***Operator*** comprises:

- ***NationalOperatorCode:*** Unique national identifier of operator. This element is to support a future planned national operator code.
- ***OperatorCode:*** Unique Identifier of operator within document.
- ***PrivateCode:*** Alternative Identifier of operator for support of legacy codes. (+TXC v2.4,)
- ***OperatorNamesGroup***: Contains elements relating to the Operator's name. See below
- ***OperatorLicenceGroup***: Contains elements relating to the Operator's Licence. See below.
- ***OtherLicences:*** *Alternative Licences for the operator.* (+TXC v2.4)
- ***OperatorParentGroup:*** Information the relationship with other parent operators. See below. (+TXC v2.4)
- ***OperatorInfoGroup:*** Further information about the operator. See below. (+TXC v2.4).

- ***OperatorContactGroup:*** Information about how to contact the operator. See below. (+TXC v2.4)
- ***AccessibilityBookings:*** Information about making booking for wheelchair and accessible use for services which the operator runs. See below. (+TXC v2.5).
- ***Garages:*** The garages which the operator runs. See below.
- ***ContributorGroup:*** Information about data rights held by operator. See below. (+TXC v2.4).

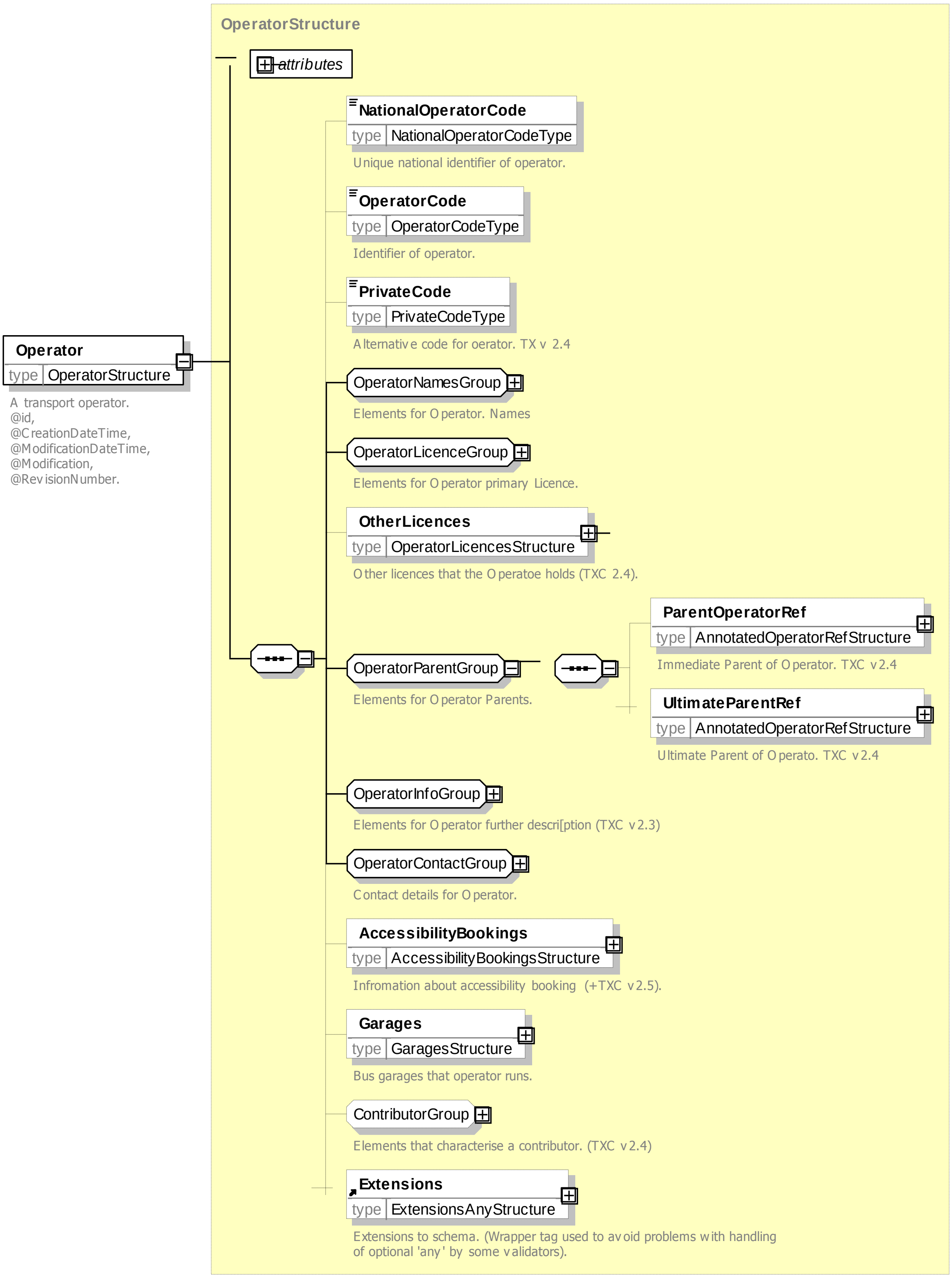


Figure 6-18 – Operator Element

6.5.3 LicensedOperator Element

The ***LicensedOperator*** element (*Figure 6-18*) is identical to the ***Operator*** element except that certain fields are mandatory.

- ***OperatorNameOnLicence, LicenceNumber, LicenceClassification***.
- ***LicensedOperatorContactGroup: ContactTelephoneNumber, EnquiryTelephoneNumber, OperatorAddresses***.

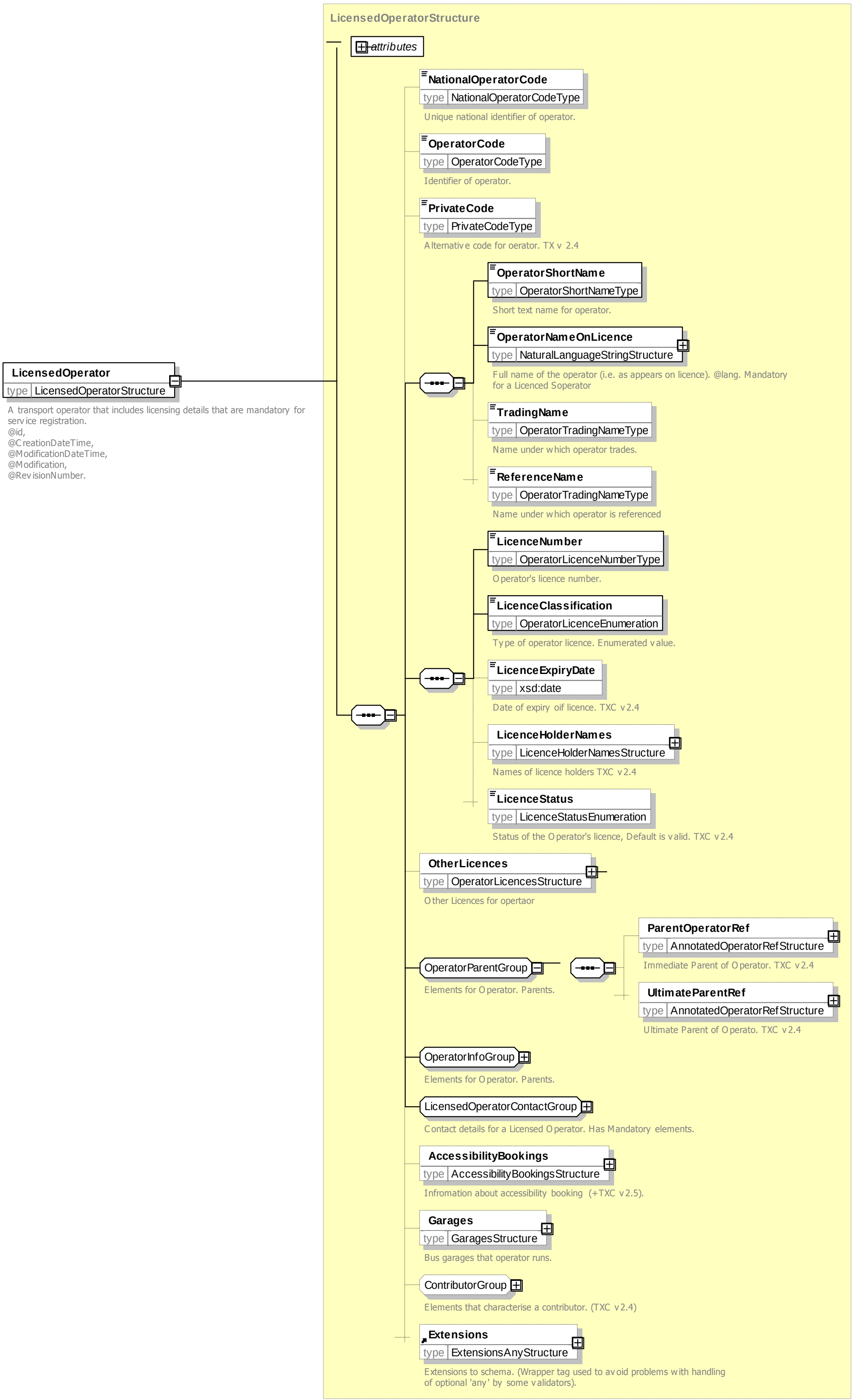


Figure 6-19 – LicensedOperator Element

6.5.4 Operator & LicensedOperator: Subelements

6.5.4.1 Operator / OperatorNamesGroup

The ***OperatorNamesGroup*** (*Figure 6-20*) describes the various names of an ***Operator*** and comprises:

- ***OperatorShortName:*** Short text name for operator.
- ***OperatorNameOnLicence:*** Full name of the operator, as it appears on licence.
- ***TradingName:*** The name under which operator trades.
- ***ReferenceName*** Name used to distinguish operator from similarly named operators. For use by data providers and managers. Not normally for public use. (+TXC v2.4).

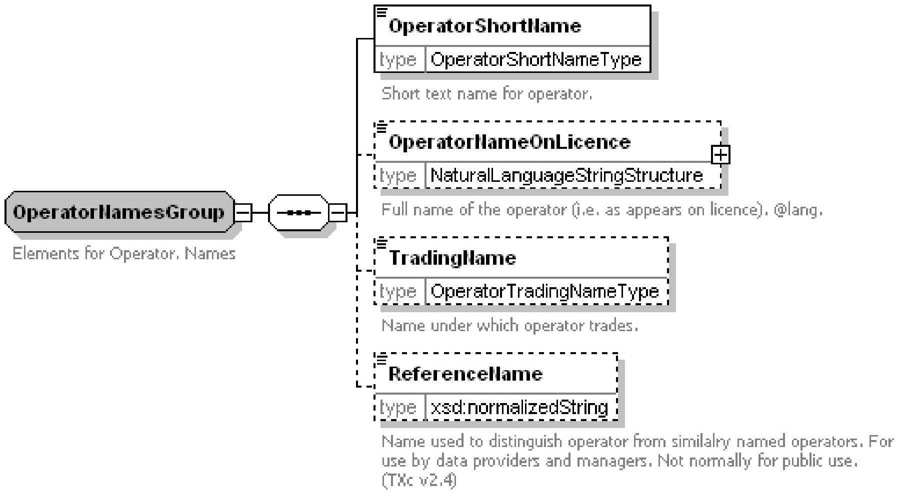


Figure 6-20 – OperatorNamesGroup

6.5.4.2 Operator / OperatorLicenceGroup

The ***OperatorLicenceGroup*** (Figure 6-21) describes the licence details for an ***Operator*** of a service and comprises:

- ***LicenceNumber:*** Operator's licence number.
- ***LicenceClassification***: Type of licence that the operator has. See Table 6-5.

Value	Description
standardNational	Standard National Licence type.
standardInternational	Standard International Licence type.
restricted	Restricted Licence type.
specialRestricted	Special Restricted Licence type.
communityBusPermit	Community Bus Permit Licence type.

Table 6-5 – Allowed Values for LicenceClassification

- ***LicenceExpiryDate:*** Date of Expiry of Operator's licence. (+TXC v2.4,)
- ***LicenceHolderNames***: Names listed as licence holders. (+TXC v2.4,)
- ***LicenceStatus:*** Status of Operator's licence. (+TXC v2.4,)

Value	Description
Valid	Valid
Refused	Refused
Surrendered	Surrendered
Continuation not sought	Continuation not sought
Revoked	Revoked
Withdrawn	Withdrawn
Application in progress.	Application in progress.

Table 6-6 – Allowed Values for LicenceStatus

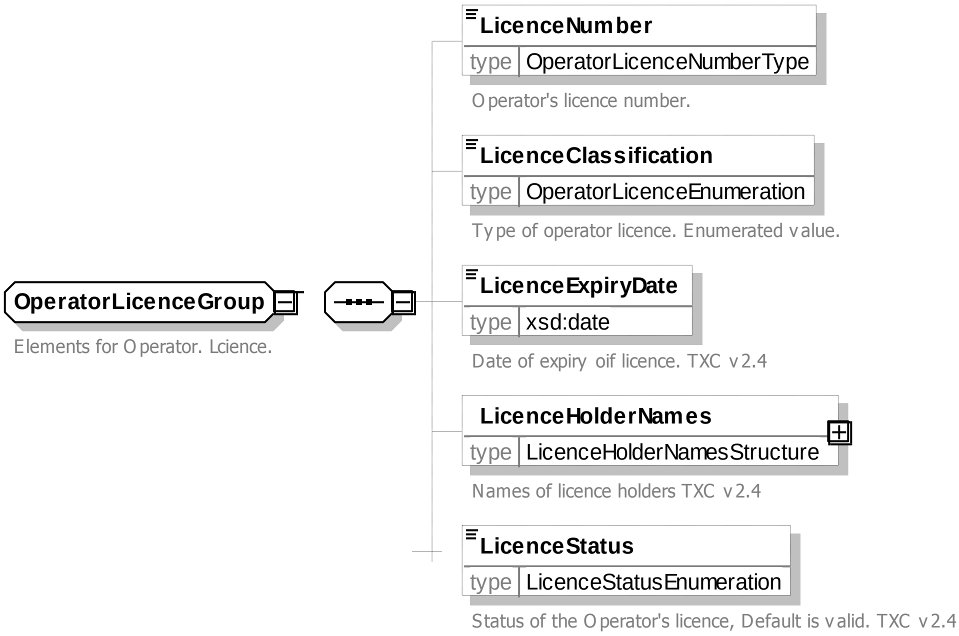


Figure 6-21 – Operator / OperatorLicenceGroup

6.5.4.3 Operator / OtherLicences Element (+TXC v2.4)

The ***Operator / OtherLicences*** (*Figure 6-22)* element additional licences that the ***Operator*** holders:

- ***OtherLicences:*** Collection of Licences. (+TXC v2.4)
    - ***OperatorLicence***: See OperatorLicenceGroup for details. (+TXC v2.4)

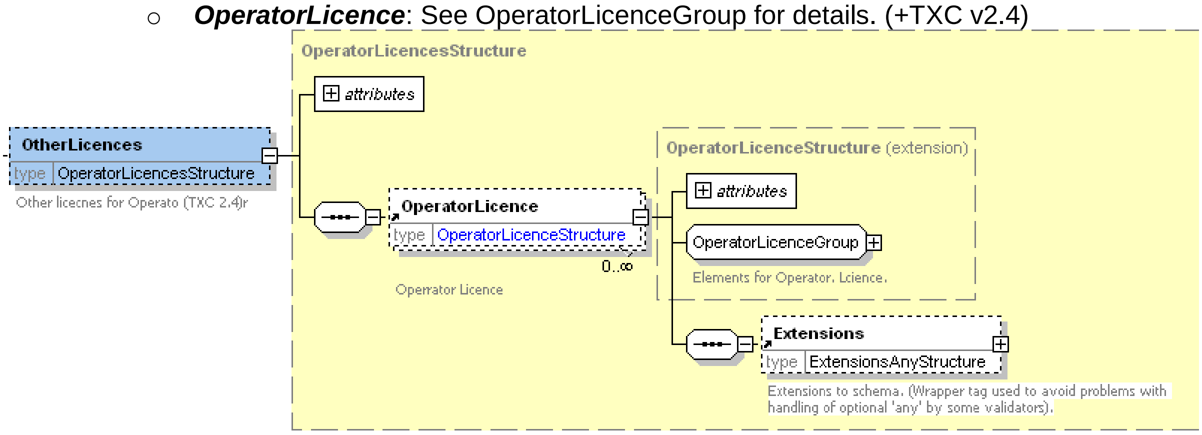


Figure 6-22 – Operator / OperatorLicence Element

6.5.4.4 Operator / OperatorParentGroup (+TXC v2.4)

The ***OperatorParentGroup*** (Figure 6-21) describes the relationships of an ***Operator*** with any parent operator and comprises:

- ***ParentOperatorRef:*** Immediate parent of Operator. (+TXC v2.4)
    - ***OperatorRef***: Identifier of Operator.
    - ***OperatorNamesGroup; See above***
- ***UltimateParentOperatorRef:*** Ultimate parent of Operator. (+TXC v2.4). As for ***ParentOperatorRef.***

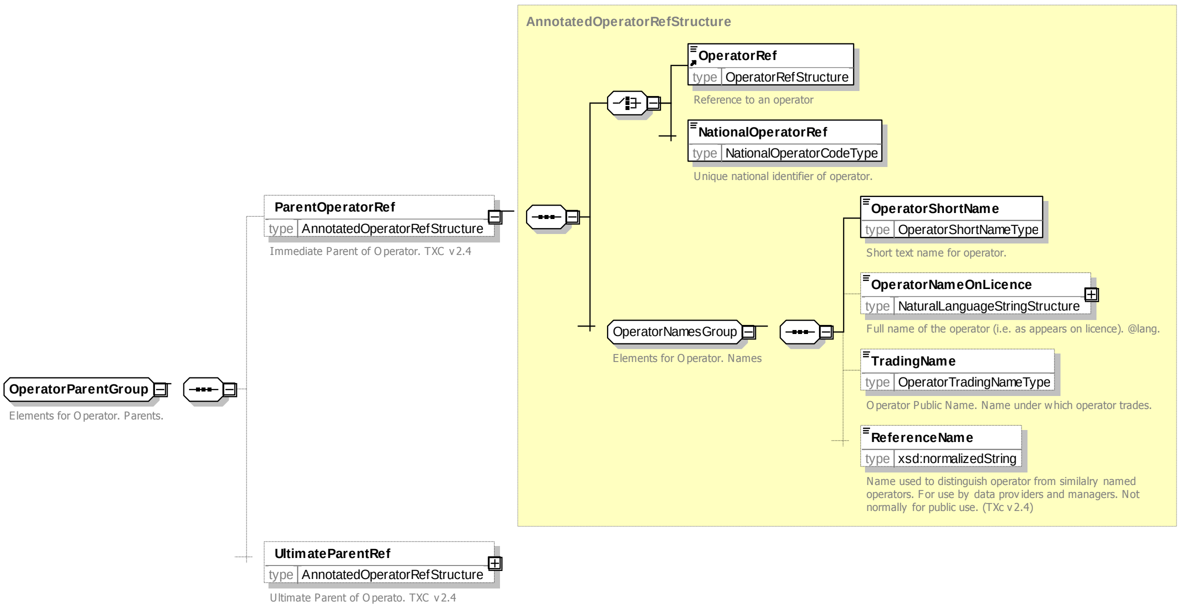


Figure 6-23 – Operator / OperatorParentGroup

6.5.4.5 Operator / OperatorInfoGroup (+TXC v2.4)

The ***OperatorInfoGroup*** (Figure 6-24) describes further details for an ***Operator*** of a service and comprises:

- ***PrimaryMode:*** The main mode the operator provides, e.g. bus, coach, etc. (+TXC v2.4).
- ***EbsrUser***: Whether Operator provides timetables to VOSA Electronic Bus Service Registration (EBSR). Default is true. (+TXC v2.4).
- ***TravelineOwner***: Primary Traveline Region who owns data record. (+TXC v2.4). Table 6-7 shows the allowed values.
- ***RegionalOperatorRefs***: Translation of operator codes to other alias for different regions. (+TXC v2.4). See below.
- ***Note***: Further comment on the operator (+TXC v2.4).

Value	Description
SW	South West
WA	Wales
YO	Yorkshire
XS	MDV combined area for London & South East
WM	West Midlands
SC	Scotland
NW	North West
NE	North East
Other	Other

Table 6-7 – Allowed Values for Data Owner

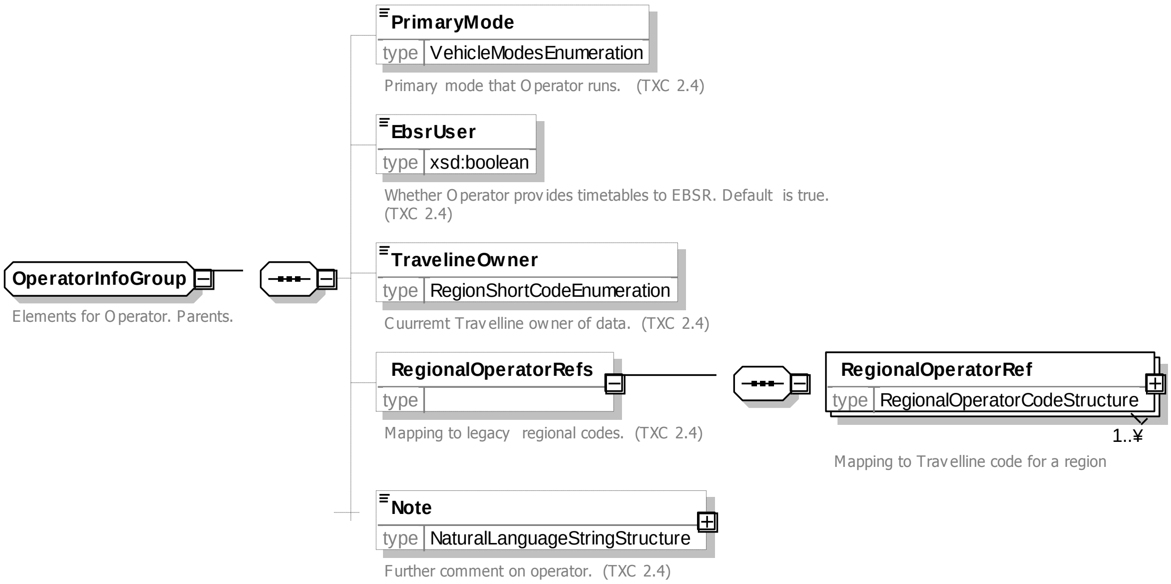


Figure 6-24 – Operator / OperatorInfoGroup

6.5.4.6 Operator / RegionalOperatorRef (+TXC v2.4)

The ***RegionalOperatorRef*** (*Figure 6-26*) maps a regional operator code to the operator:

- ***TravelineRegion:*** Traveline Region in which code is used (+TXC v2.4). Table 6-7 above shows the allowed values.
- ***TravelineOperatorCode:*** Code used in specified Traveline Region.

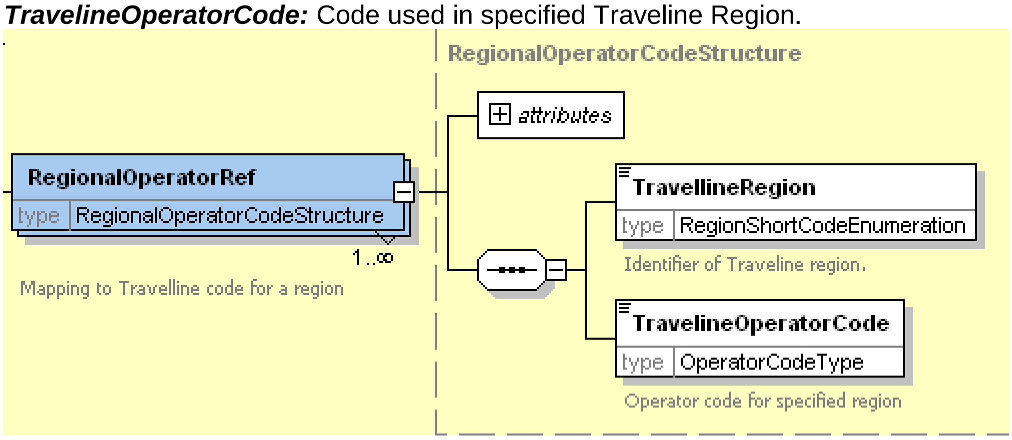


Figure 6-25 – Operator / RegionalOperatorRef

6.5.4.7 Operator / OperatorContactGroup

The ***OperatorContactGroup*** (*Figure 6-26*) describes the contact details for an ***Operator*** of a service and comprises:

- ***EnquiryTelephoneNumber:*** Telephone Number for public enquiries to the operator concerning the service. See ***TelephoneContactStructure*** in common schema elements in *Section 7*.
- ***ContactTelephoneNumber:*** Telephone Number to contact operator concerning the service. See ***TelephoneContactStructure*** below.
- ***ContactFaxNumber:*** Fax Number to contact operator concerning the service. See ***TelephoneContactStructure*** below.
- ***ContactPerson:*** Name of contact person.
- ***CustomerServiceTelephoneNumber:*** *General customer Service* Telephone Number to contact operator See ***TelephoneContactStructure*** below.

- ***OperatorAddresses:*** Operator's addresses. A separate ***OperatorAddress*** and ***CorrespondanceAddress*** can be specified. See ***PostalAddressStructure*** in Common Schema Elements in *Section 7*.
- ***EmailAddress:*** The email address of the operator. It is up to the operator whether an individual's address or a generic company e-mail address is used.
- ***WebSiteAddress:*** The web site URL of the operator.

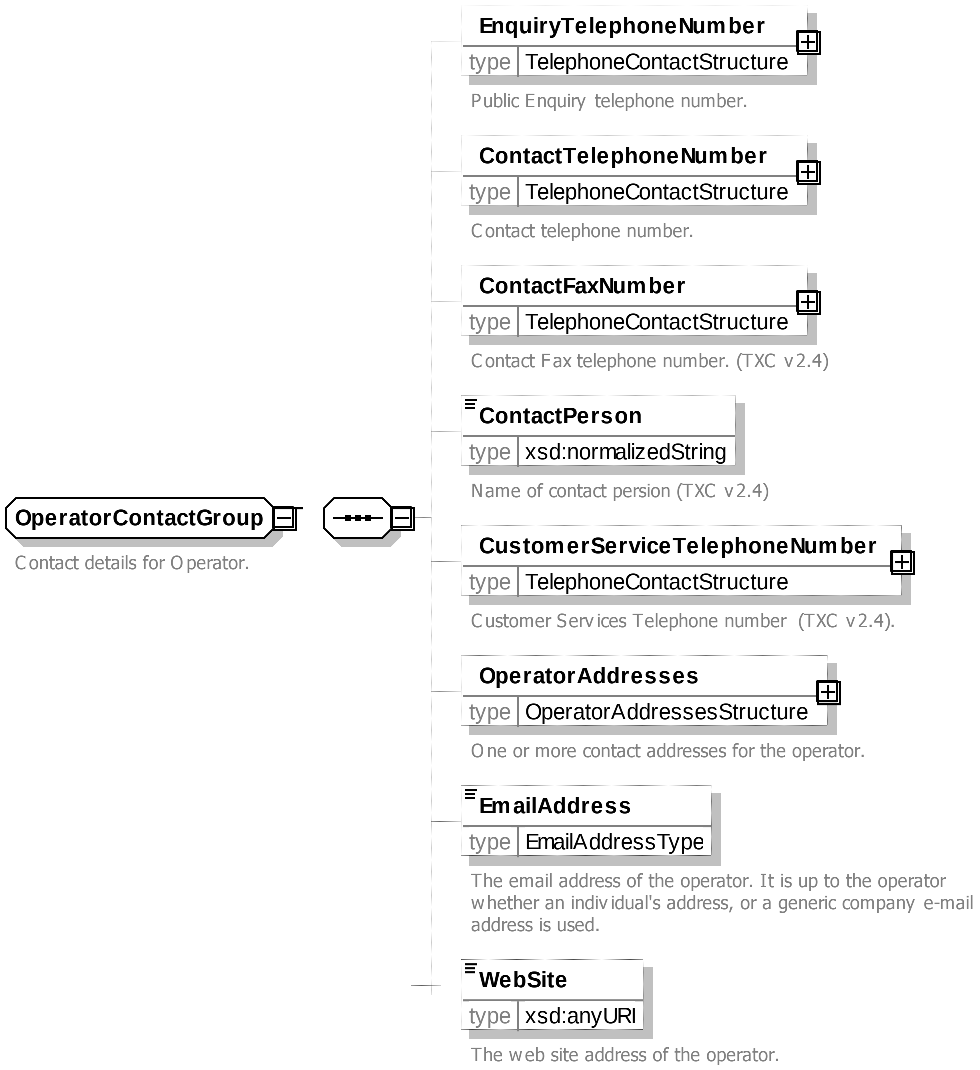


Figure 6-26 – Operator / OperatorContactGroup

6.5.4.8 Operator / AccessibilityBooking / (+TXC v2.5)

The ***AccessibilityBooking*** element records details about how to book wheelchair use of an operator's services (Figure 6-29).

- ***Mode:*** Transport mode of operator's service to which this information relates.
- ***Regions:*** Travel line regions to which to which this information relates for an operator.
  - ***RegionRef*** reference to a Traveline region.
- ***AssistanceAvailability:*** Availability of services for wheelchair users (Table 6-8).shows the allowed values for assistance availability.

Value	Description
none	Assistance service is not available from Operator.
available	Assistance is available from Operator.
availableIfBooked	Assistance is available if booked.
availableAtCertainTimes	Assistance is available at certain times.
unknown	Not known if available.

Table 6-8 – Allowed Values for AssistanceAvailability

- ***ContactTelephoneNumber:*** Telephone Number to contact operator concerning accessibility. See ***TelephoneContactStructure*** below.
- ***EmailAddress:*** Email address to contact operator concerning accessibility.
- ***BookingMethod:*** Methods of booking allowed. See Table 6-12 for allowed values.

• Value	Description
online	Book Online
callDriver	Book by phoning driver
callOffice	Book by phoning Office
phonesAtStop	Book using phone at stop
text	Book by sending text
none	No method
other	Other Booking Method

Table 6-9 – Allowed Values for BookingMethod

- ***MinimumBookingPeriod:*** Minimum interval in advance of departure day or time that service may be ordered.
- ***LatestBookingTime***: Latest time in the day that a booking can be made.
- ***BookingUrl***: The web site address for accessibility info and or booking.
- ***Note***: Additional note on booking arrangements.

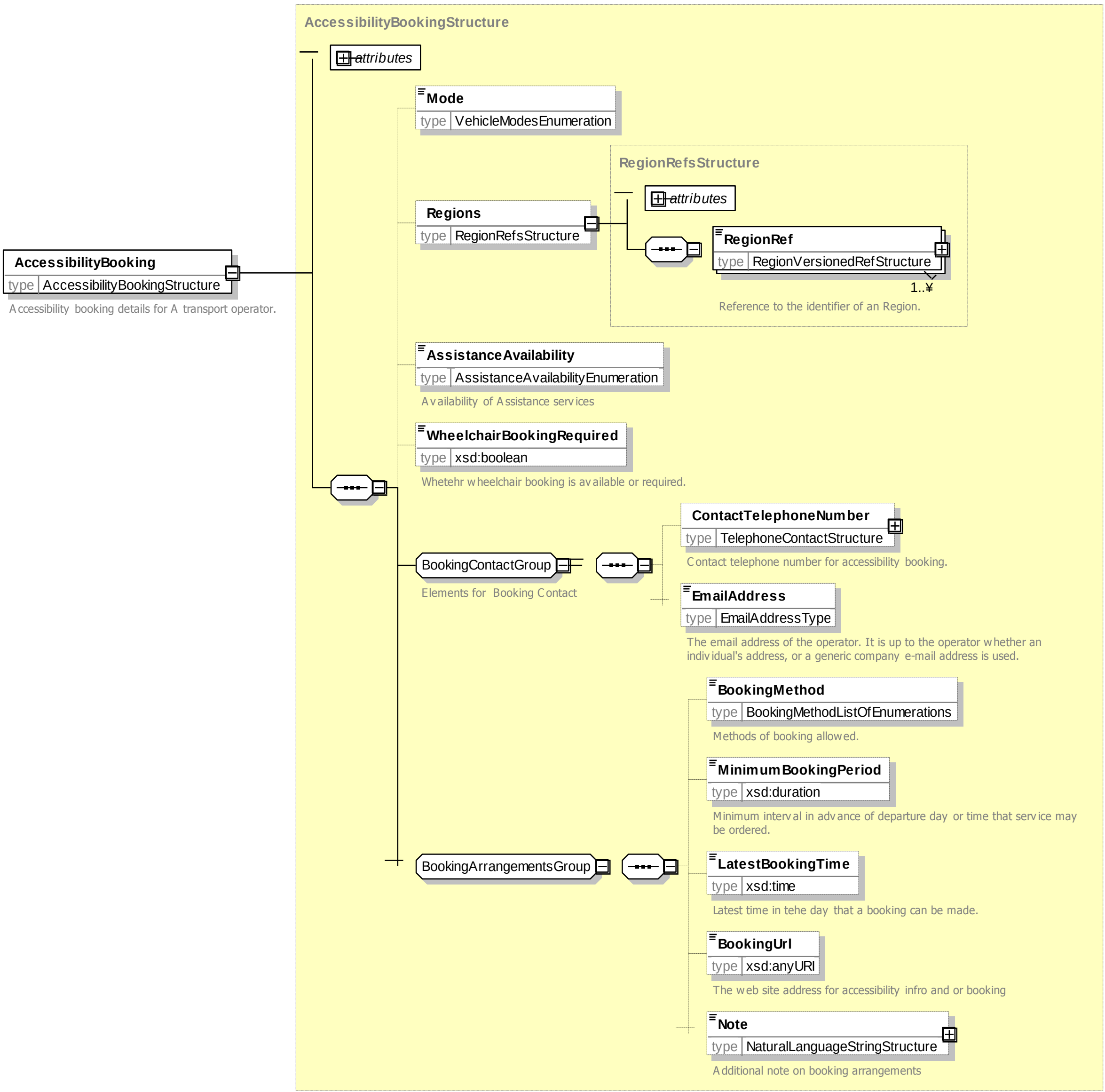


Figure 6-27 – Operator / AccessibilityBooking element

6.5.4.9 Operator / Garages Element

The ***Operator / Garages*** element records details about the garages or depots which the operator uses. It contains a collection of ***Garage*** (*Figure 6-28*) elements.
Each ***Garage*** is composed of:

- ***GarageCode***: Identifier of garage. This will be referenced by other elements.
- ***GarageName***: Name of garage.
- ***ContactNumber:*** Telephone Number to contact for queries about operational data. See ***TelephoneContactStructure*** in Common Schema Elements in *Section 7*.
- ***Address***: Postal Address of garage. See ***PostalAddressStructure*** in Common Schema Elements in *Section 7*.
- ***Location***: Spatial coordinates of garage.

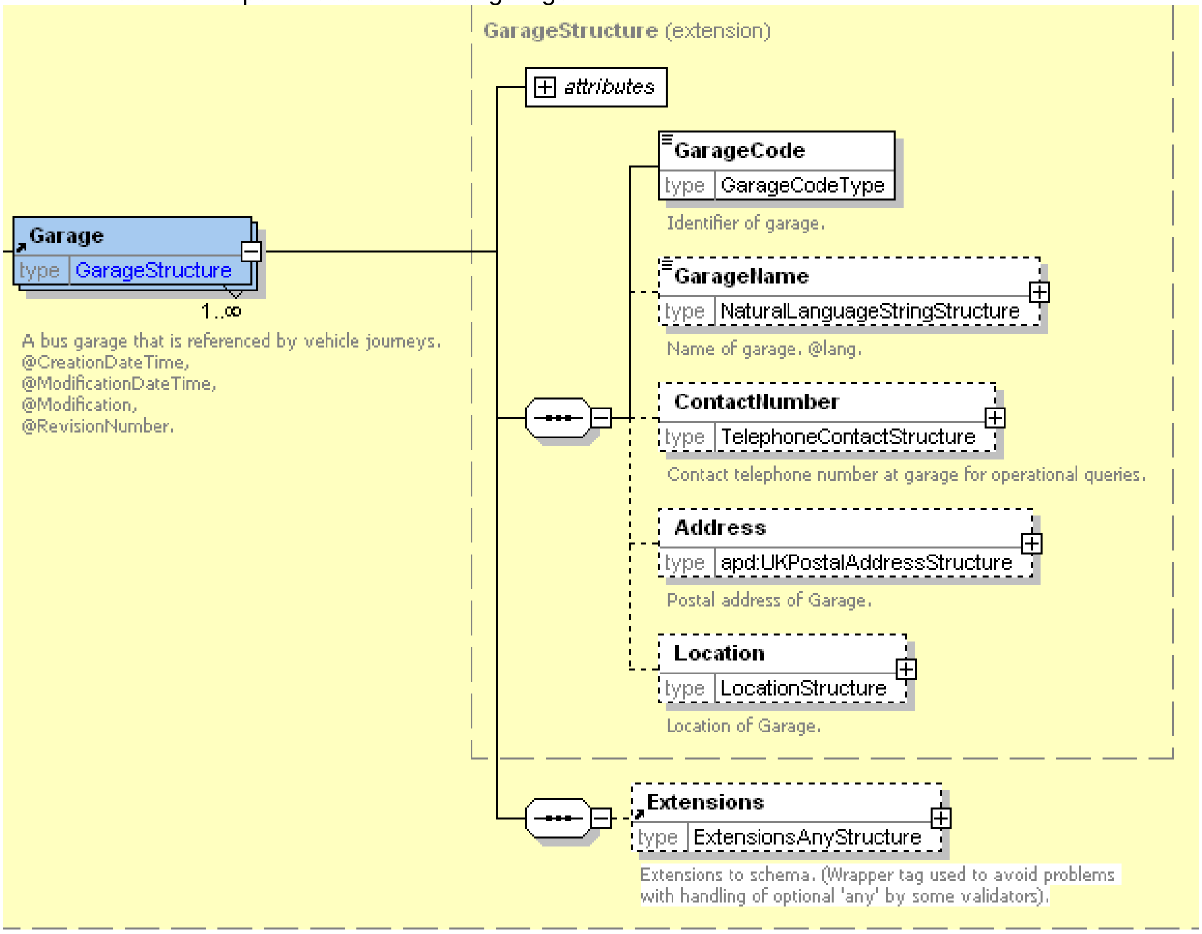


Figure 6-28 – Operator / Garages / Garage Element

6.5.4.10 Operator / ContributorGroup (+TXC v2.4)

The ***ContributorGroup*** element records details about the data rights management for the operator. (Figure 6-29):

- ***PolicyStatus***: Status under the Freedom of Information Act. (TXC v2.4). See Table 6-10 for allowed values. Defaults to *ExemptFromFreedomOfInformation*. If not specified, use value of any containing context

Value	Description
SubjectToFreedomOfInformation	Holder is Subject to the Freedom Of information Act
ExemptFromFreedomOfInformation	Holder is not Subject to the Freedom Of information Act
Other	Other status

Table 6-10 – Allowed Values for Policy Status

- ***ContributorClassification***: Type of Contributor. (TXC v2.4). See Table 6-11 for allowed values.

Value	Description
CommunityMember	Contributor is a private individual
ProfessionalOrganisation	Contributor is a professional Organisation
LocalAuthority	Contributor is a local authority.
ProfessionalOrganisationForLocalAuthority	Contributor is a professional Organisation acting on behalf of a Local Authority.

Table 6-11 – Allowed Values for Contributor Classification

- ***DataRights:*** One or more Data rights offered by Organisation. See below.

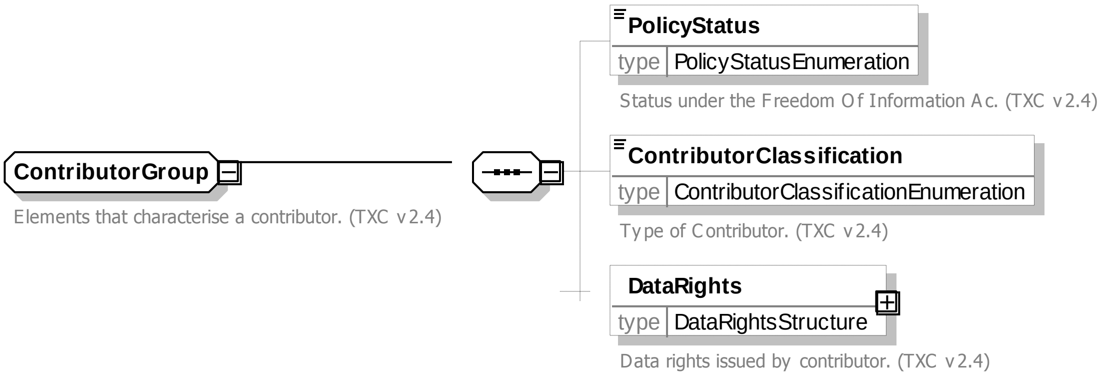


Figure 6-29 – Operator / ContributorGroup Element

6.5.4.11 DataRight / (+TXC v2.4)

The ***DataRight*** element records details about a data rights (Figure 6-29):

- ***TermsOfUse*** elements:
- ***AllowedUse***: Permitted use of data element or elements. (TXC v2.4). See Table 6-12 for allowed values.

Value	Description
CommercialLicence	Data may be used under a commercial licence
OpenLicence	Data may be used under an Open Source Licence
NotForProfit	Data has may be used for free under an Open Source Licence provided service does not itself charge user for s use.
Unrestricted	Data may be used without charge, subject to terms and conditions.

Table 6-12 – Allowed Values for AllowedUse

  - ***TermsUrl***: Reference to web URL with terms and conditions of use. (TXC v2.4).
  - ***TermsAndConditions***: Text for terms and conditions of use. (TXC v2.4).
- ***Copyright*** elements:
  - ***CopyrightUrl***: Reference to web URL with copyright statement. (TXC v2.4).
  - ***CopyrightStatement*** Text for copyright. (TXC v2.4).
- ***Policy*** elements:
- ***PolicyStatus***: Policy status of element covered by right. (TXC v2.4). See Table 6-13for allowed values.

Value	Description
SubjectToFreedomOfInformationt	Element is Subject to the Freedom Of information Act
ExemptFromFreedomOfInformation	Element is not Subject to the Freedom Of information Act
Other	Other status

Table 6-13 – Allowed Values for PolicyStatus

  - ***PolicyUrl***: URL providing information on the policy. (TXC v2.4).
  - ***PolicyJustification*** Text providing information on the policy (TXC v2.4).
- ***DataRights:*** One or more Data rights offered by Organisation. See below.

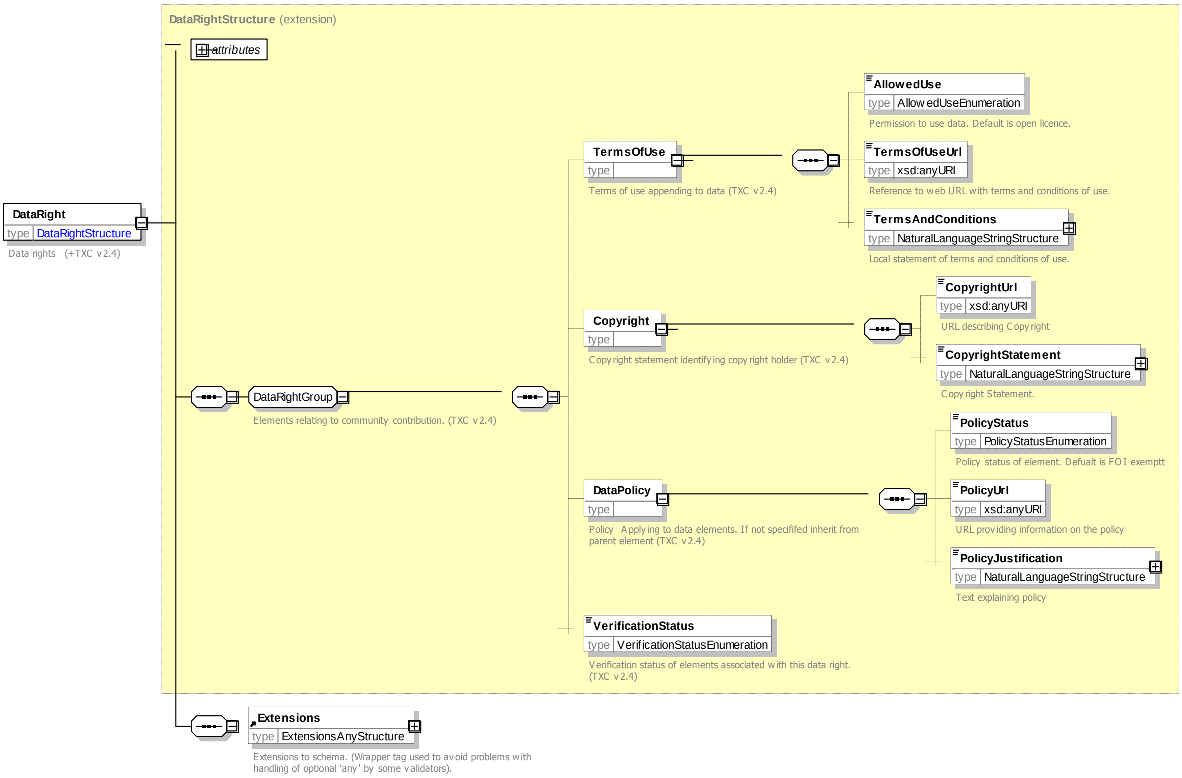


Figure 6-30 – Operator / DataRight Element

6.5.5 Registration Element

The ***Registration*** element (*Figure 6-31*) records statutory administrative details about the registration of the service. In the *TransXChange Registration Schema* the element is mandatory; in the *TransXChange General Schema* it is not. A ***Registration*** comprises:

- ***ServiceRef***: The ***Service*** that the registration covers.
- ***RegistrationSubmissionGroup***: Describes basic properties of registration.
- ***RegistrationInfoGroup***: Describes further properties of the registration.
- ***ShortNoticeRegistration:*** Additional information to support a registration made with less than the statutory period of notice. See later below.

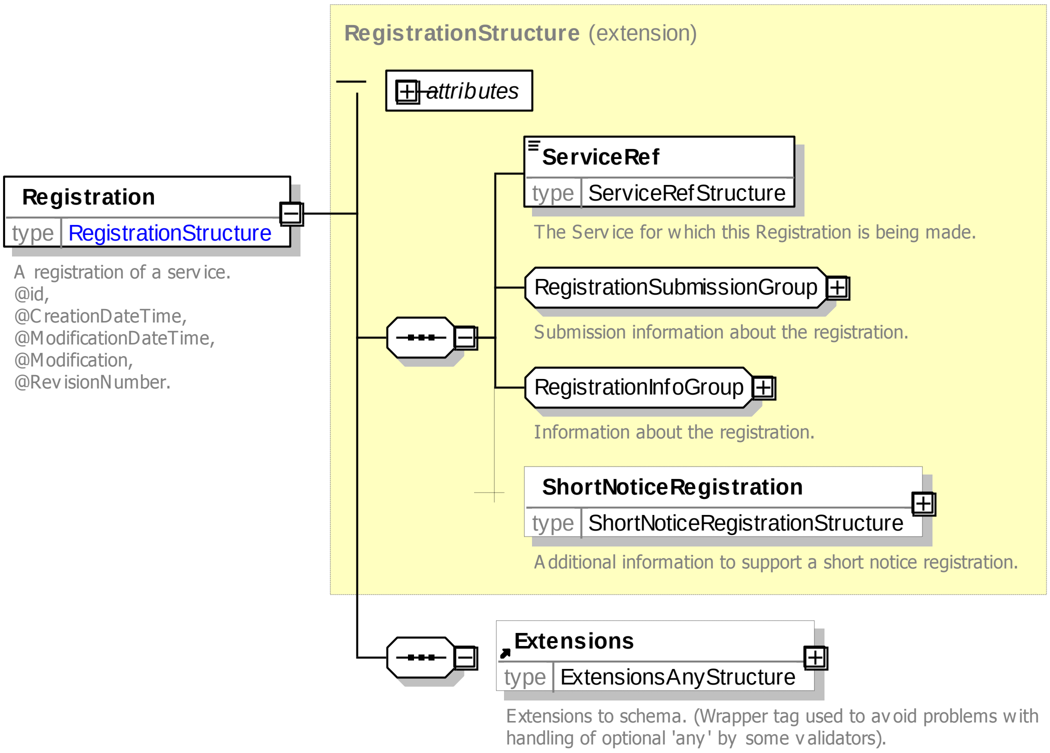


Figure 6-31 – Registration Element

6.5.6 Registration / RegistrationSubmissionGroup

The ***RegistrationSubmissionGroup*** (*Figure 6-32*) holds elements describing the basic submission of registration.

- ***SubmissionDate***: Intended date of ***Registration*** submission by submitter (officially received date may be different).
- ***VosaRegistrationNumber***: The identifiers for the ***Registration***. See below.
- ***RegistrationWorkflowGroup:*** Elements concerning the processing of the Registration. See below.
- ***VariationNumber***: Variation number of the registration.
- ***SubmissionAuthor***: Contact details of person submitting registration. See below.
- ***EbsrAgent***: Name of Agency if registration is prepared and submitted by a proxy
- ***TrafficAreas:*** A collection of ***TrafficArea*** instances with full or partial responsibility for the registration of the submission. See below.
- ***CirculatedAuthorities:*** Collection of ***CirculatedAuthority*** instances to whom the registration is to be circulated. See below.

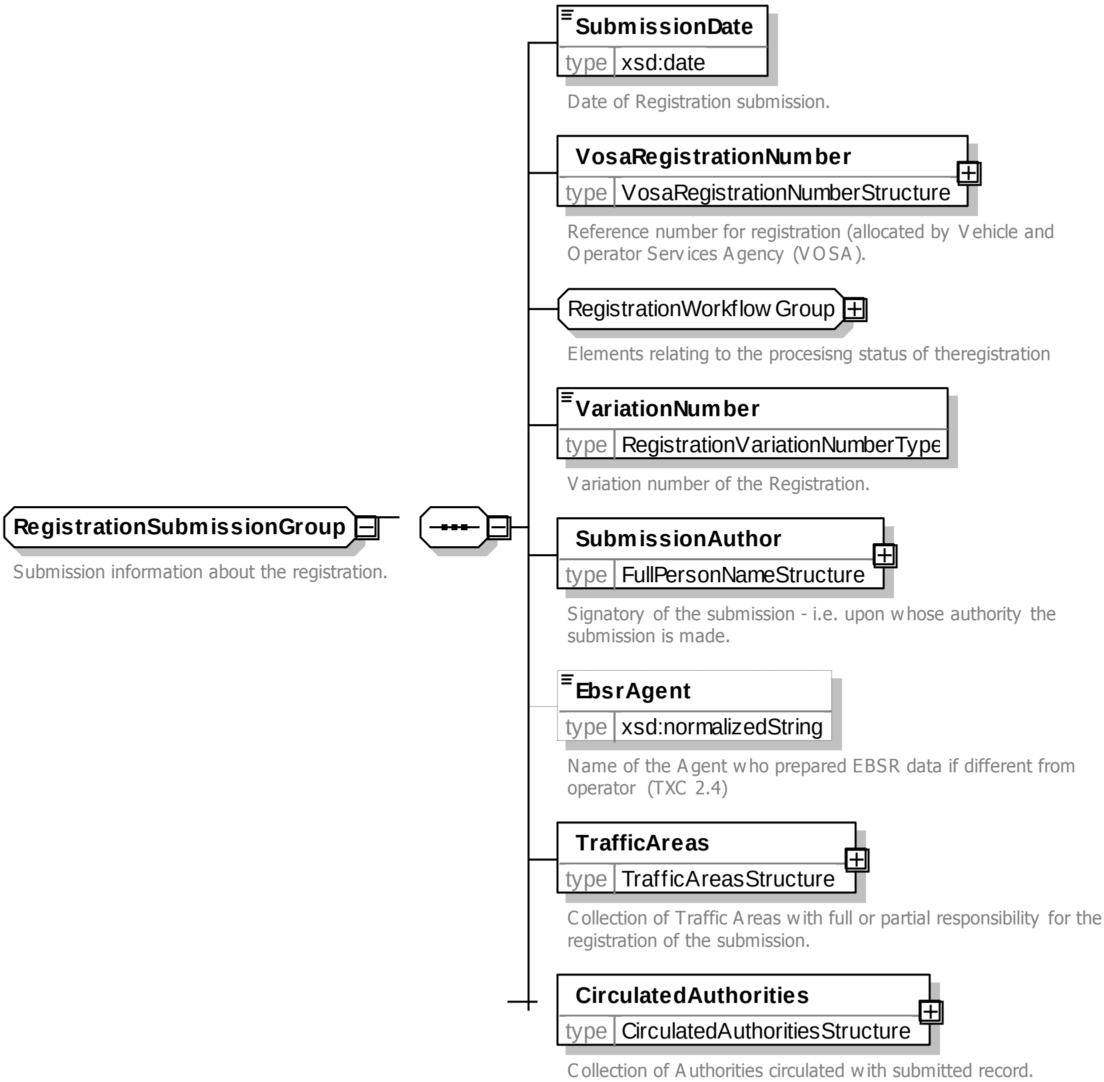


Figure 6-32 – RegistrationSubmissionGroup

6.5.7 Registration / RegistrationWorkflowGroup

The ***RegistrationWorkflowGroup*** (*Figure 6-32*) holds elements describing the basic submission of registration.

- ***ApplicationClassification***: Type of the registration application. See Table 6-14*.*

Value	Description
new	New registration.
chargeableChange	Chargeable modification of an existing registration.
nonChargeableChange	Non-chargeable modification of an existing registration.
cancel	Cancellation of a registration.

Table 6-14 – Allowed Values for Registration / ApplicationClassification

- ***RegistrationWorkflowStatus***: Current processing status of the application. See Table 6-15. Allows tracking of current status of a submission. *(+TXC 2.4*)

Value	Description
draft	Document is a working preapproval draft
submitted	Document is a submitted schedule not yet approved
rejected	Document has been rejected
elaboratedAfterAcceptance	Document is an accepted schedule that has been further augmented with details.
underConsultation	Document is a submitted proposal that has been acknowledged by VOSA but is not yet approved

accepted	Document is an accepted schedule
variant	Document is an issued variant that does not need approval
other	Document has some other status – e.g. for a non-registration

Table 6-15 – Allowed Values for Registration / RegistrationWorkflowStatus

- ***RegistrationVariantNature***: Nature of non-statutory variant effect. Indicates nature of change i.e. how document will affect downstream systems. See Table 6-16*.* *(+TXC 2.**4***)

Value	Description
all	Document changes all or most aspects of registration
footnotes	Document has changes to notes that does not affect schedule
journeys	Document has changes to add or remove journeys that will affect schedule
journeyPattern	Document has changes to alter stops of journeys that will affect schedule
other	Change is not ascribed to any section
operational	Document has changes to operational data that does not affect schedule
profile	Document has changes to availability that may affect schedule
registration	Document has changes to registration particulars
textual	Document has only textual changes
timings	Document has changes to timings of journeys that will affect schedule
track	Document has changes to route track data that does not affect schedule

Table 6-16 – Allowed Values for Registration / RegistrationVariantNature

- ***RegistrationSubVariantNumber***: Subnumber identifying Non-statutory change to a previously submitted registration. Should be unique**.** ***(+TXC 2.4***).
- ***StatutoryChange*** Whether change affects formal registration*.* ***(+TXC 2.4***). Default is *true.*

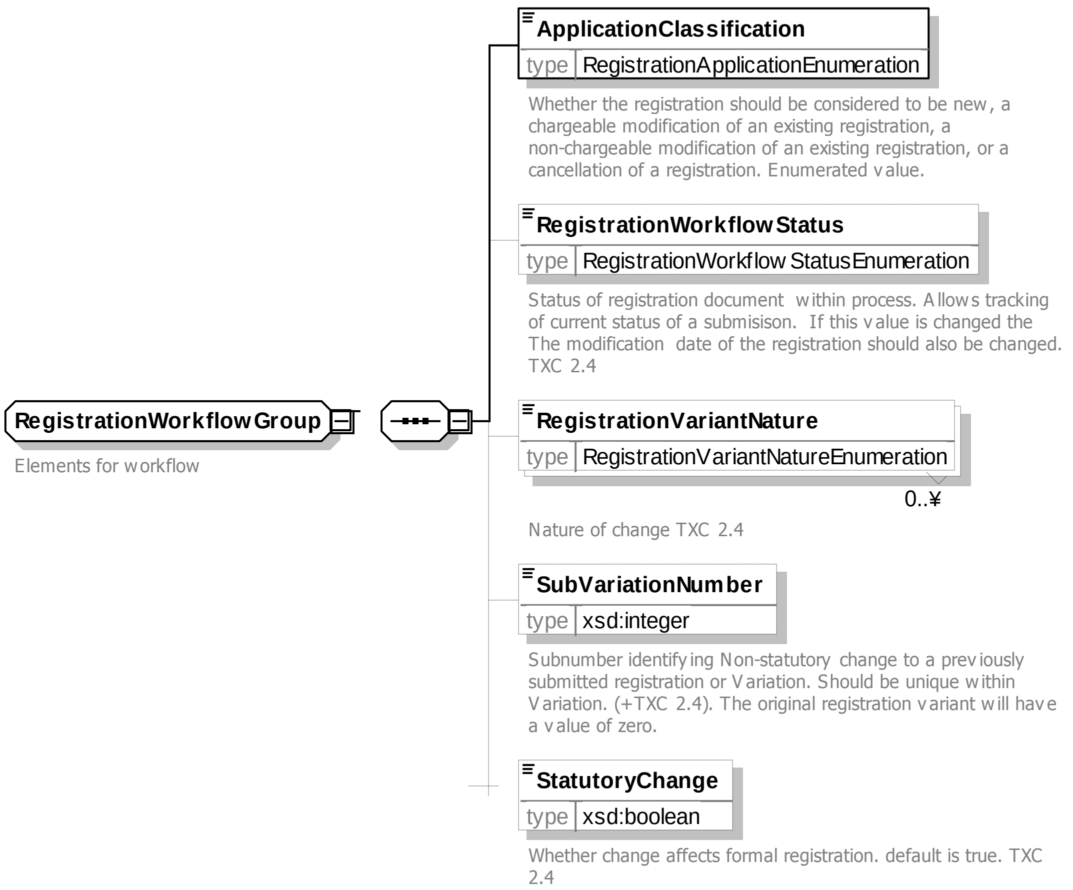


Figure 6-33 – Registration Workflow Group

6.5.8 Registration / RegistrationInfoGroup

The ***RegistrationInfoGroup*** (*Figure 6-34*) holds elements describing additional properties of a registration.

- ***SubsidyDetails:*** Information about any subsidy of the ***Service***. See below.

- ***ContractedService:*** Information about any contract under which the ***Service*** is run for an authority. See below.
- ***QualityPartnership:*** Information about any Statutory Quality partnership under which the ***Service*** is run.
- ***SupportingDocuments:*** Names of additional documents that accompany the registration. Note that references to any schematic maps that are in image format should be placed with the ***Service*** / ***SchematicMap*** element, and not here.

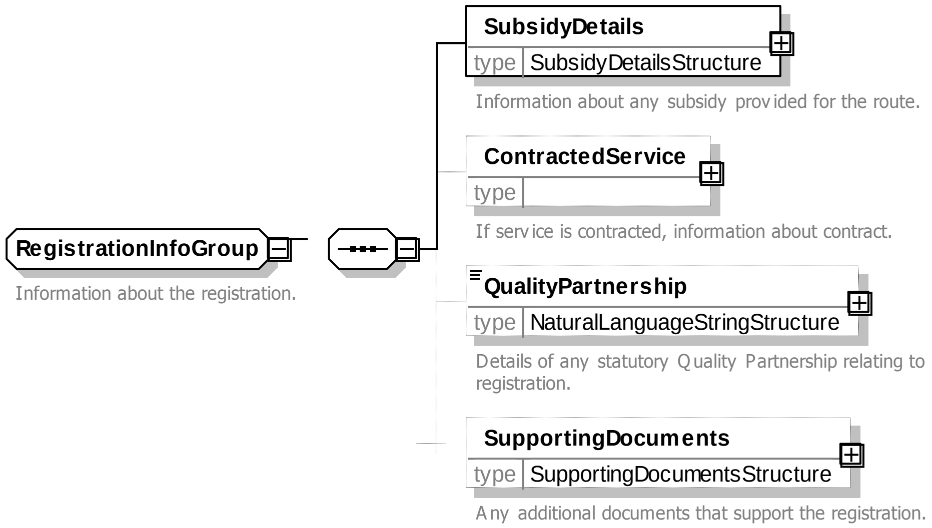


Figure 6-34 – RegistrationInfoGroup

6.5.9 Registration Subelements

6.5.9.1 Registration / VosaRegistrationNumber Element

The ***VosaRegistrationNumber*** element (*Figure 6-35*) specifies the unique identifiers of the ***Registration***. It is made up of three components:

- ***TanCode:*** Two character Traffic Area prefix. See *Table 6-17.*

Value	Description
PB	North Eastern Traffic Area
PC	North Western Traffic Area
PD	West Midlands Traffic Area
PF	Eastern Traffic Area
PG	Welsh Traffic Area
PH	Western Traffic Area
PK	South Eastern and Metropolitan Traffic Area
PM	Scottish Traffic Area

Table 6-17 – Allowed Values for TanCode

- ***LicenceNumber:*** The Registered operator's seven character licence number. This should be the same as the ***Operator / LicenceNumber*** value.
- ***RegistrationNumber***: Unique identifier of registration for licence holder. 1-4 numeric only characters.

When displayed, numbers include a separator slash between the licence number and the suffix, for example 'PB1235601/456'.

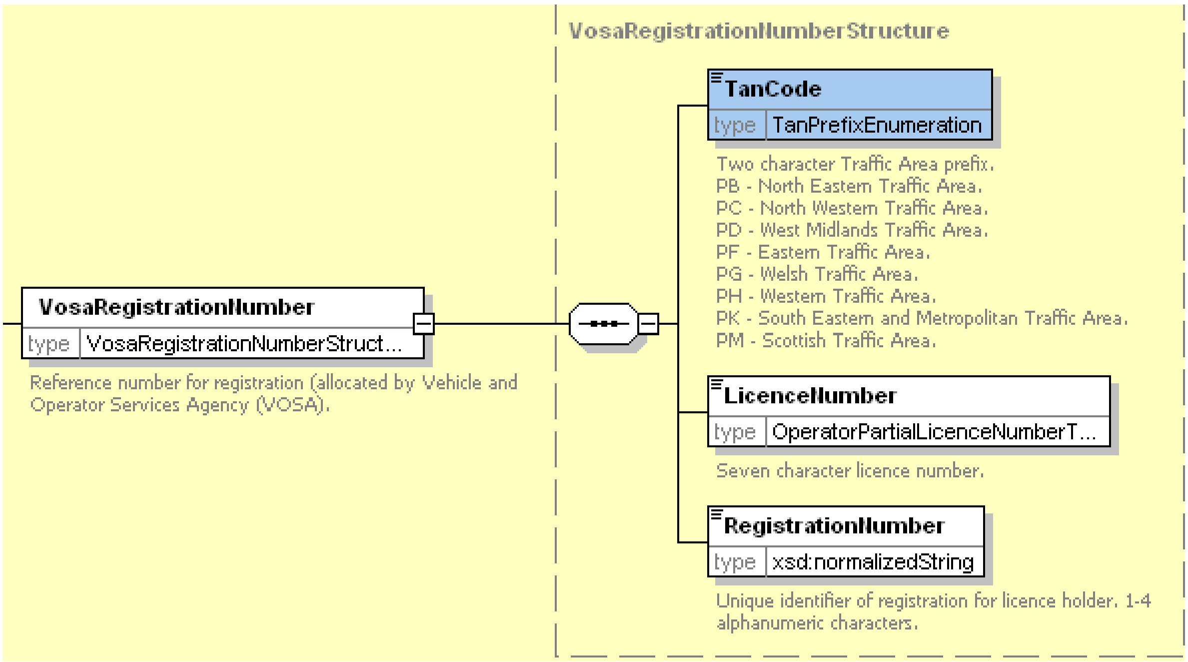


Figure 6-35 – Registration / VosaRegistrationNumber Element

6.5.9.2 Registration / SubmissionAuthor Element

The ***SubmissionAuthor*** (*Figure 6-36*) describes the signatory of the submission – that is, upon whose authority the submission is made. It comprises:

- ***Position:*** Position of the signatory of the ***Registration***.
- ***Title:*** Title of the signatory of the ***Registration***.
- ***Forename:*** Forename of the signatory of the ***Registration***.
- ***Surname:*** Surname of the signatory of the ***Registration***.

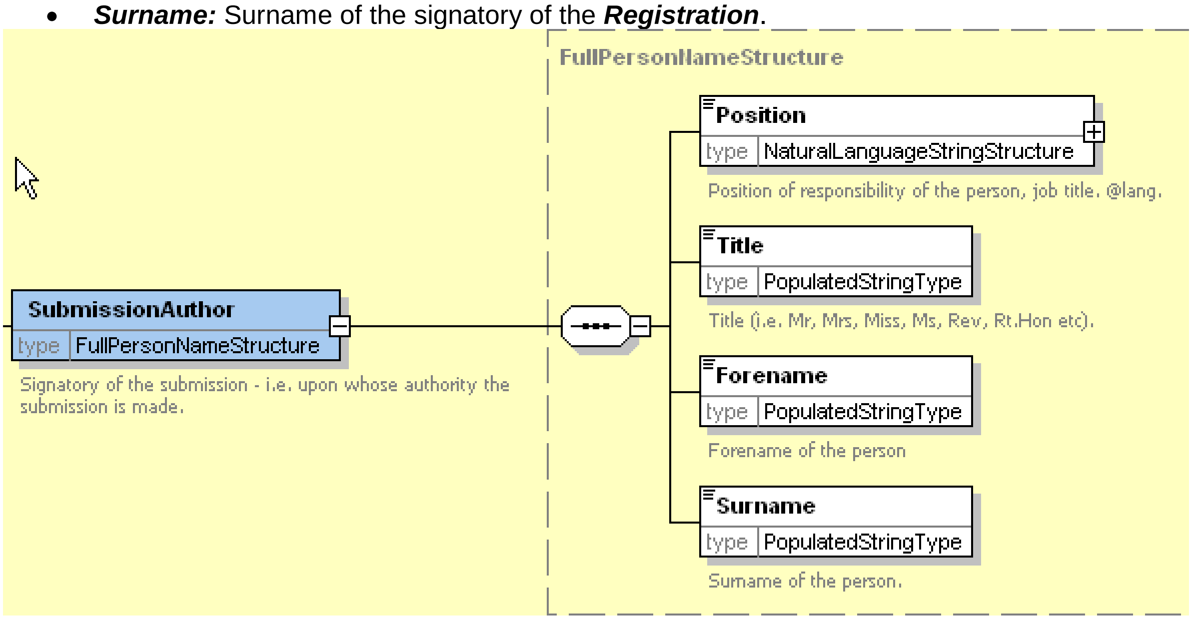


Figure 6-36 – Registration / SubmissionAuthor Element

6.5.9.3 Registration / TrafficArea Element

The ***TrafficAreas*** element (*Figure 6-37*) lists the individual ***TrafficArea*** elements for the registration.

- ***TrafficAreaName***: Specifies a TrafficArea – see *Table 6-18*.

Value	**Description**
Eastern	Eastern.

NorthEastern	North Eastern.
NorthWestern	North Western.
SouthEastMetropolitan	South East Metropolitan.
Scottish	Scottish.
Welsh	Welsh.
WestMidlands	West Midlands.
Western	Western.

Table 6-18 – Allowed Values for TrafficArea / Names

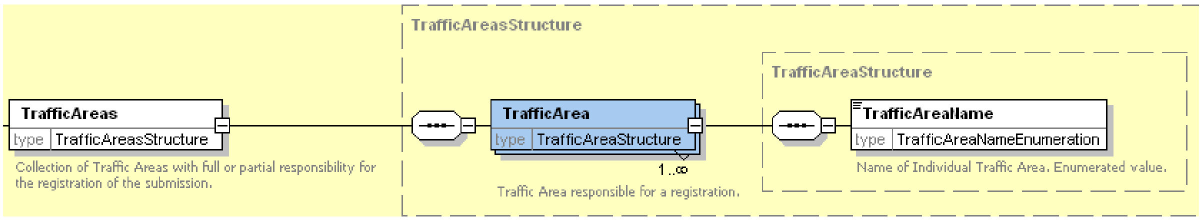


Figure 6-37 – Registration / TrafficArea Element

6.5.9.4 Registration / CirculatedAuthorities Element

The ***CirculatedAuthorities*** element (*Figure 6-38*) lists the individual ***CirculatedAuthority*** elements for the registration.

- ***CirculatedAuthority:*** Names identifying circulated authority. May be specified in one of two ways:
  - ***AuthorityName:*** Name of circulated authority from validated list. Should be used if Authority is in current list. See *Table 6-19.*
  - ***UnverifiedAuthorityName:*** Unvalidated name of circulated authority. For other use. (TXC v2.4). Definitive name for new areas should be obtained from VOSA.

Value	**English**	**TXC version**
Aberdeen	Aberdeen	2.0
Aberdeenshire	Aberdeenshire	2.0
Angus	Angus	2.0
ArgyllAndBute	Argyll and Bute	2.0
BathAndNorthEastSomerset	Bath and North East Somerset	2.0
Bedford	Bedford	2.2
CentralBedfordshire	*CentralBedfordshire*	2.2
Bedfordshire	Bedfordshire	2.0 – Deprecated 2.3
Berkshire	Berkshire	2.0
BlackburnWithDarwen	Blackburn with Darwen	2.0
Blackpool	Blackpool	2.0
BlaenauGwent	Blaenau Gwent	2.0
Bournemouth	Bournemouth	2.0
BracknellForest	Bracknell Forest	2.0
Bridgend	Bridgend	2.0
BrightonAndHove	Brighton and Hove	2.0
Bristol	Bristol	2.0
Buckinghamshire	Buckinghamshire	2.0
Caerphilly	Caerphilly	2.0
Cambridgeshire	Cambridgeshire	2.0

Cardiff	Cardiff	2.0
Carmarthenshire	Carmarthenshire	2.0
CentroWestMidlands	Centro (West Midlands)	2.0
Ceredigion	Ceredigion	2.0
ChannelIslands	Channel Islands	2.0
Cheshire	Cheshire	2.0 Deprecated 2.3
CheshireEast	Cheshire East	2.2
CheshireWestAndChester	Cheshire West and Chester	2.2
Clackmannanshire	Clackmannanshire	2.0
ComhairleNanEileanSiar	Comhairle Nan Eilean Siar	2.0
Conwy	Conwy	2.0
CornwallAndScillies	Cornwall and Scillies	2.0
Cumbria	Cumbria	2.0
Darlington	Darlington	2.0
Denbighshire	Denbighshire	2.0
Derby	Derby	2.0
Derbyshire	Derbyshire	2.0
Devon	Devon	2.0
Dorset	Dorset	2.0
DumfriesAndGalloway	Dumfries and Galloway	2.0
Dundee	Dundee	2.0
Durham	Durham	2.0
EastAyrshire	East Ayrshire	2.0
EastDunbartonshire	East Dunbartonshire	2.0
EastLothian	East Lothian	2.0
EastRenfrewshire	East Renfrewshire	2.0
EastRidingOfYorkshire	East Riding of Yorkshire	2.0
EastSussex	East Sussex	2.0
Edinburgh	Edinburgh	2.0
Essex	Essex	2.0
ExeterCity	Exeter City	2.4
Falkirk	Falkirk	2.0
Fife	Fife	2.0
Flintshire	Flintshire	2.0
Glasgow	Glasgow	2.0
Gloucestershire	Gloucestershire	2.0
GMPTE	GMPTE (Manchester)	2.0
Gwynedd	Gwynedd	2.0
Halton	Halton	2.0
Hampshire	Hampshire	2.0
Hartlepool	Hartlepool	2.0

Havering	Havering	2.0
Herefordshire	Herefordshire	2.0
Hertfordshire	Hertfordshire	2.0
Highland	Highland	2.0
Inverclyde	Inverclyde	2.0
IsleOfAnglesey	Isle of Anglesey	2.0
IsleOfMan	Isle of Man	2.0
IsleOfWight	Isle of Wight	2.0
Kent	Kent	2.0
KingstonUponHull	Kingston Upon Hull	2.0
Lancashire	Lancashire	2.0
Leicester	Leicester	2.0
Leicestershire	Leicestershire	2.0
Lincolnshire	Lincolnshire	2.0
London	London	2.0
Luton	Luton	2.0
Medway	Medway	2.0
Merseytravel	Merseytravel	2.0
MerthyrTydfil	Merthyr Tydfil	2.0
MetroWestYorks	Metro (West Yorks)	2.0
Middlesbrough	Middlesbrough	2.0
Midlothian	Midlothian	2.0
MiltonKeynes	Milton Keynes	2.0
Monmouthshire	Monmouthshire	2.0
Moray	Moray	2.0
NeathPortTalbot	Neath Port Talbot	2.0
WestBerkshire	West Berkshire	2.0
Newport	Newport	2.0
NexusTyneside	Nexus (Tyneside)	2.0
Norfolk	Norfolk	2.0
NorthAyrshire	North Ayrshire	2.0
NorthEastLincolnshire	North East Lincolnshire	2.0
NorthernIreland	Northern Ireland	2.0
NorthLanarkshire	North Lanarkshire	2.0
NorthLincolnshire	North Lincolnshire	2.0
NorthSomerset	North Somerset	2.0
NorthYorkshire	North Yorkshire	2.0
Northamptonshire	Northamptonshire	2.0
Northumberland	Northumberland	2.0
NorwichCity	NorwichCity	2.4
Nottingham	Nottingham	2.0
Nottinghamshire	Nottinghamshire	2.0

OrkneyIslands	Orkney Islands	2.0
Oxfordshire	Oxfordshire	2.0
Pembrokeshire	Pembrokeshire	2.0
PerthAndKinross	Perth and Kinross	2.0
Peterborough	Peterborough	2.0
Plymouth	Plymouth	2.0
Poole	Poole	2.0
Portsmouth	Portsmouth	2.0
Powys	Powys	2.0
Reading	Reading	2.0
RedcarAndCleveland	Redcar and Cleveland	2.0
Renfrewshire	Renfrewshire	2.0
RhonddaCynonTaff	Rhondda Cynon Taff	2.0
Rutland	Rutland	2.0
ScottishBorders	Scottish Borders	2.0
ShetlandIslands	Shetland Islands	2.0
Shropshire	Shropshire	2.0
Slough	Slough	2.0
Somerset	Somerset	2.0
SouthAyrshire	South Ayrshire	2.0
SouthGloucestershire	South Gloucestershire	2.0
SouthLanarkshire	South Lanarkshire	2.0
SouthYorkshirePTE	South Yorkshire PTE	2.0
Southampton	Southampton	2.0
SouthendOnSea	Southend On Sea	2.0
Staffordshire	Staffordshire	2.0
Stirling	Stirling	2.0
StocktonOnTees	Stockton On Tees	2.0
StokeOnTrent	Stoke On Trent	2.0
StrathclydePTE	Strathclyde PTE	2.0
Suffolk	Suffolk	2.0
Surrey	Surrey	2.0
Swansea	Swansea	2.0
Swindon	Swindon	2.0
TelfordAndWrekin	Telford and Wrekin	2.0
Thurrock	Thurrock	2.0
Torbay	Torbay	2.0
Torfaen	Torfaen	2.0
ValeOfGlamorgan	Vale of Glamorgan	2.0
Warrington	Warrington	2.0
Warwickshire	Warwickshire	2.0
WestDunbartonshire	West Dunbartonshire	2.0

WestLothian	West Lothian	2.0
WestSussex	West Sussex	2.0
Wiltshire	Wiltshire	2.0
WindsorAndMaidenhead	Windsor and Maidenhead	2.0
Wokingham	Wokingham	2.0
Worcestershire	Worcestershire	2.0
York	York	2.0

Table 6-19 – Allowed Values for CirculatedAuthority Names

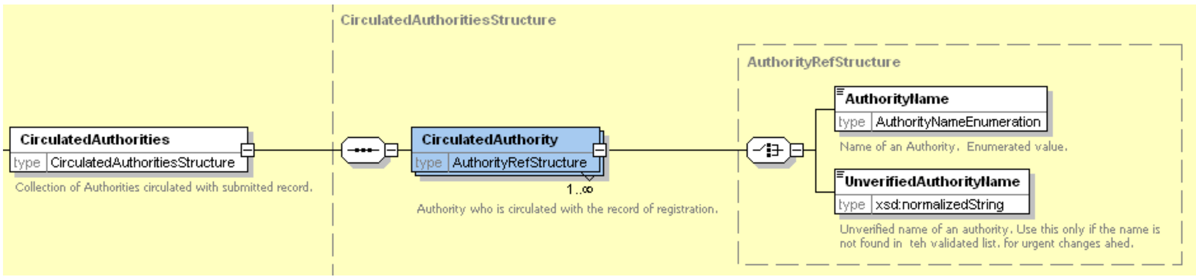


Figure 6-38 – Registration / CirculatedAuthorities Element

6.5.9.5 Registration / SubsidyDetails Element

The ***SubsidyDetails*** element (*Figure 6-39*) gives information about any subsidy that applies to the ***Registration***.
Either there are none – ***NoSubsidy***, or there is a ***Subsidy,*** made up of two elements:

- ***SubsidyType:*** Whether subsidy is full or partial. *Table 6-20*.

Value	Description
partial	Partial subsidy applies.
full	Full subsidy applies.

Table 6-20 – Allowed Values for SubsidyType

- ***SubsidisingAuthority:*** Name of subsidising authority.

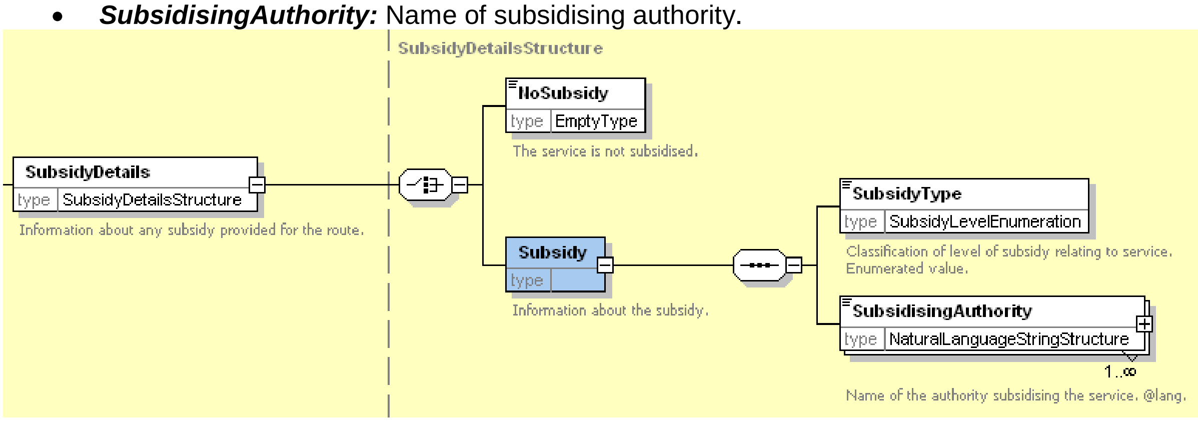


Figure 6-39 – Registration / SubsidyDetails Element

6.5.9.6 Registration / ContractedService Element

The ***ContractedService*** element (*Figure 6-40*) specifies if the service is run under contract to a Local Authority or SPT. This item is specific to Scottish registration.
Nature of Contract:

- ***NotContracted:*** Service is not run under contract.
- ***WhollyContracted:*** Service is run wholly under contract.
- ***PartContracted:*** Service is run in part under contract.

- ***ContractingAuthority:*** Names of one or more authorities awarding contract. See ***CirculatedAuthority / AuthorityName***.

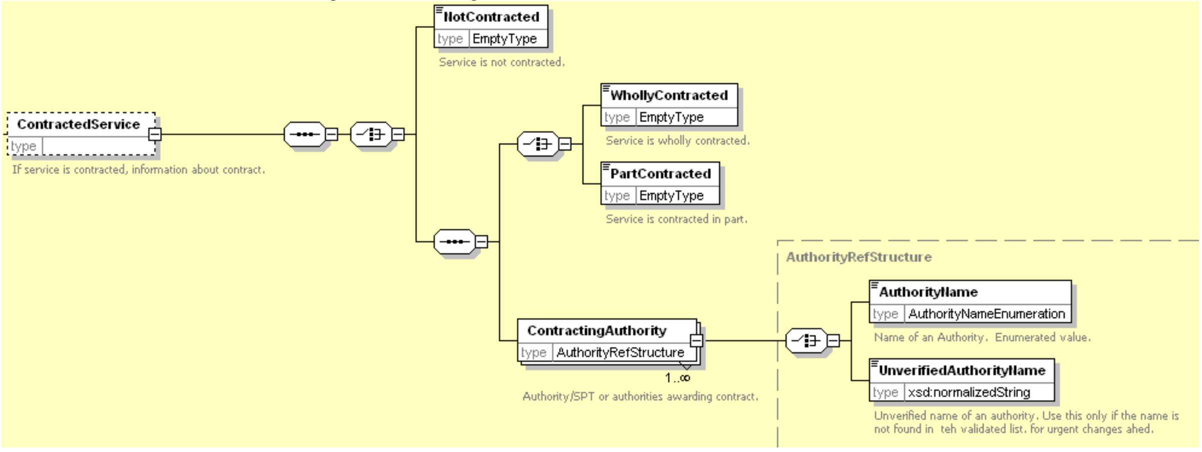


Figure 6-40 – Registration / ContractedService Element

6.5.9.7 Registration / SupportingDocument Element

The ***SupportingDocument*** element (*Figure 6-41*) Associates any supporting documents associated with the registration. Documents are identified by a ***DocumentUri***.

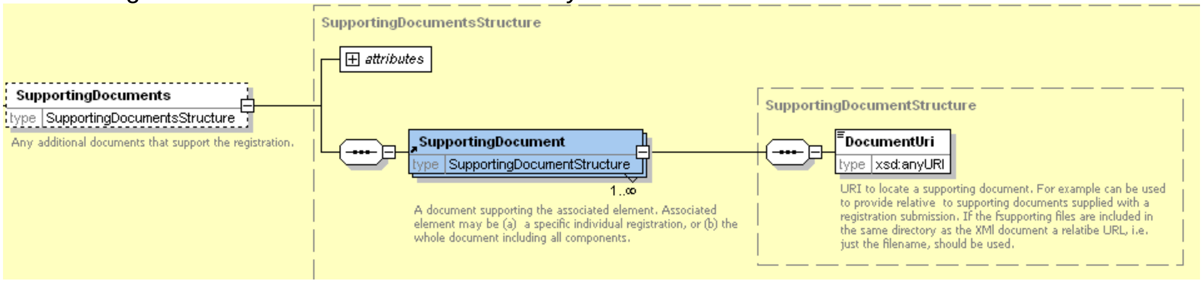


Figure 6-41 – Registration / SupportingDocument Element

6.5.10 ShortNoticeRegistration Element

A short notice registration is an application to register, cancel or change a service made with less than the normally 56 days' period of notice. Only certain determined cases can be submitted within the reduced period. A ***ShortNoticeRegistration*** requires additional details as specified by one or more elements in the ***ChangeImpactGroup*** & ***ChangeJustificationGroup***.

- ***ChangeImpactGroup***: Elements describing the impact of the change.
- ***ChangeJustificationGroup:*** Elements describing the justification(s) for the change.

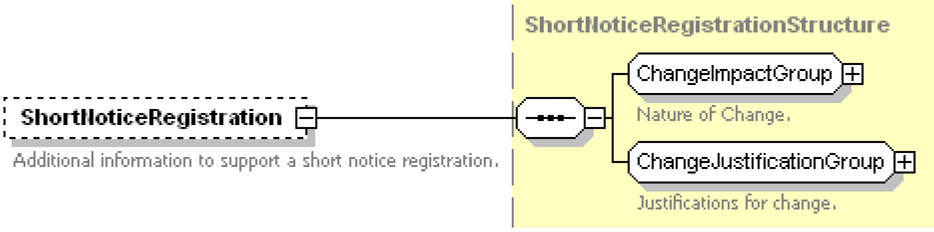


Figure 6-42 – ShortNoticeRegistration Element

6.5.11 ShortNoticeRegistration / ChangeImpactGroup

The ***ChangeImpactGroup*** (*Figure* 6-43) holds elements describing the impact of the change. These include:

- ***PublicAvailability***: Whether the service is to be available to the general public. See below.
- ***ChangeImpact***: Whether the change to the service time is in excess of the normal allowed limits and so requires additional justification. See below.

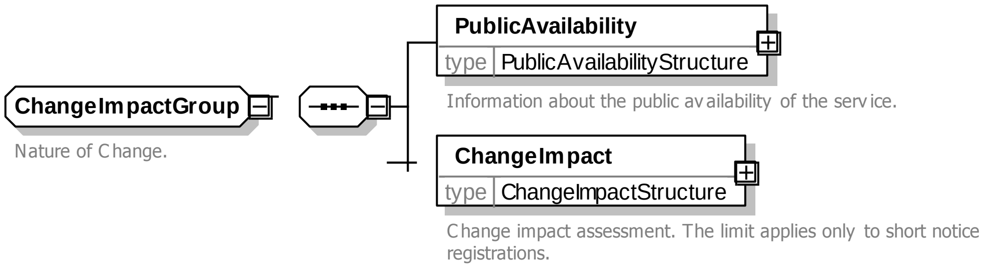


Figure 6-43 – ShortNoticeRegistration / ChangeImpactGroup

6.5.12 ShortNoticeRegistration / ChangeJustificationGroup

The ***ChangeJustificationGroup*** (*Figure 6-44*) holds elements describing the justification(s) for the change. These include:

- ***BankHolidayChange***: Whether the ***ShortNoticeRegistration*** is needed to address a bank holiday requirement. See below.
- ***ChangeToConnectAlteredService***: Whether the short notice registration is needed to handle a modification to another service. See below.
- ***ReplaceDiscontinuedService***: Whether the service is to replace a discontinued service, whose discontinuation justifies the short notice registration? See below.
- ***LocalHolidayChange***: Whether the short notice registration is to accommodate a local holiday. See below.
- ***SpecialOccasion***: Whether the short notice registration is to accommodate a special occasion. See below.
- ***RegulationOrderCompliance***: Whether the short notice registration is needed to meet a road traffic order. See below.
- ***ChangeRequestedByExternalAuthority***: Whether the short notice registration is needed to meet a request by an external authority such as the Police. See below.
- ***ExceptionalRequirement***: Whether the short notice registration is needed to meet an allowed exceptional requirement. See below.

- ***MiscellaneousJustification***: The reasons justifying the short notice registration submission where none of the above considerations are applicable. More than one reason may be included.

Figure 6-44 – ShortNoticeRegistration / ChangeJustificationGroup

6.5.13 ShortNoticeRegistration Subelements

6.5.13.1 ShortNoticeRegistration / Public Availability Element

The ***PublicAvailability*** element (*Figure 6-45*) specifies whether the service is to be available to the general public.

- ***AvailableToPublic***: Specifies service is available.
- ***NotAvailableToPublic***. Specifies service is not available, accompanied by a ***NonAvailabilityDescription.***

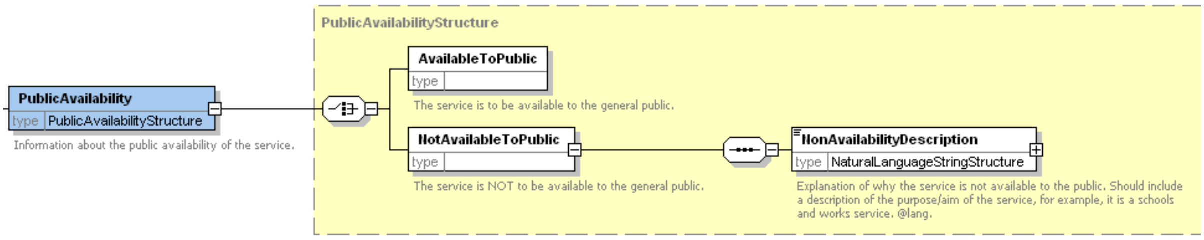


Figure 6-45 – ShortNoticeRegistration / PublicAvailability Element

6.5.13.2 ShortNoticeRegistration / ChangeImpact Element

The ***ChangeImpact*** element (*Figure 6-46*) specifies whether the change to the service time is in excess of the normal allowed limit (i.e. more than ten minutes from the current time): if the change is more than the allowed amount, then a justification must be given, otherwise a Minor Change Description can be used.

- ***ChangeExceedsLimit***: Change exceeds the allowed limit. Only possible if change to existing application, i.e. if ***ChangeClassification*** is ***Change*** or ***Cancel.***
- ***ChangeDoesNotExceedLimit:*** The change does not exceed the limit.

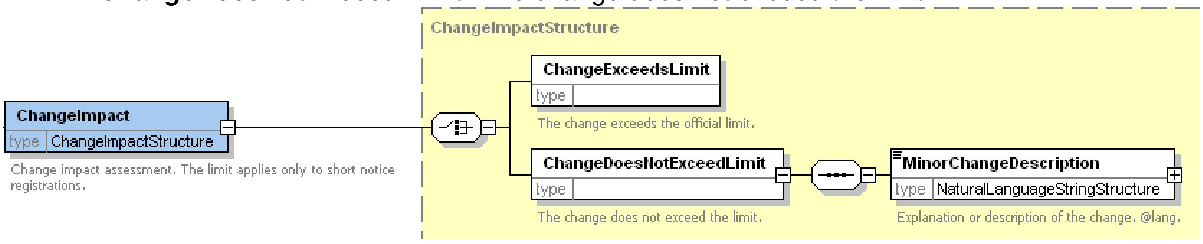


Figure 6-46 – ShortNoticeRegistration / ChangeImpact Element

6.5.13.3 ShortNoticeRegistration / ChangeToConnectAlteredService Element

The ***ChangeToConnectAlteredService*** (Figure *6-47*) specifies whether the short notice registration is needed to handle a modification to another service, and if so, which one:
It contains an ***AlteredServiceRequiringConnection*** instance, which is an ***AnnotatedServiceRefStructure.***

- ***ServiceRef:*** Reference to another ***Service*** definition provided elsewhere in the document.
- ***Description:*** Text description of the service &/or its identifier if not defined by a service reference.

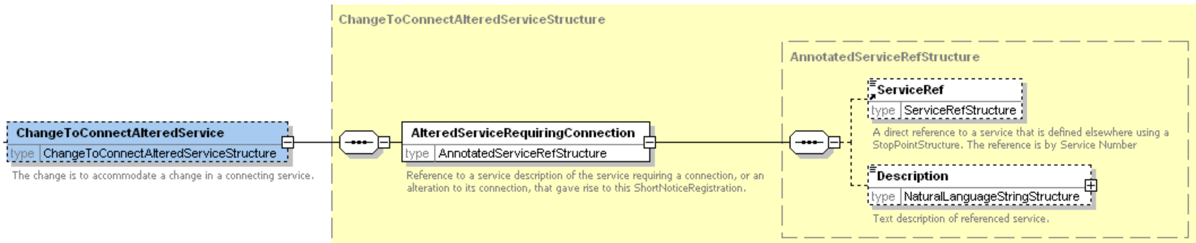


Figure 6-47 – ShortNoticeRegistration / ChangeToConnectAlteredService Element

6.5.13.4 ShortNoticeRegistration / ReplaceDiscontinuedService Element

The ***ReplaceDiscontinuedService*** (*Figure 6-48*) identifies the discontinued service which the service of the short notice registration replaces.

- ***DiscontinuedServiceOperator:*** Operator of the discontinued service.

DiscontinuedService: Description of the discontinued service, an ***AnnotatedServiceRefStructure.***

- ***ServiceRef:*** Reference to another ***Service*** definition provided elsewhere in the document.
- ***Description:*** Text description of the service &/or its identifier if not defined by a service reference.

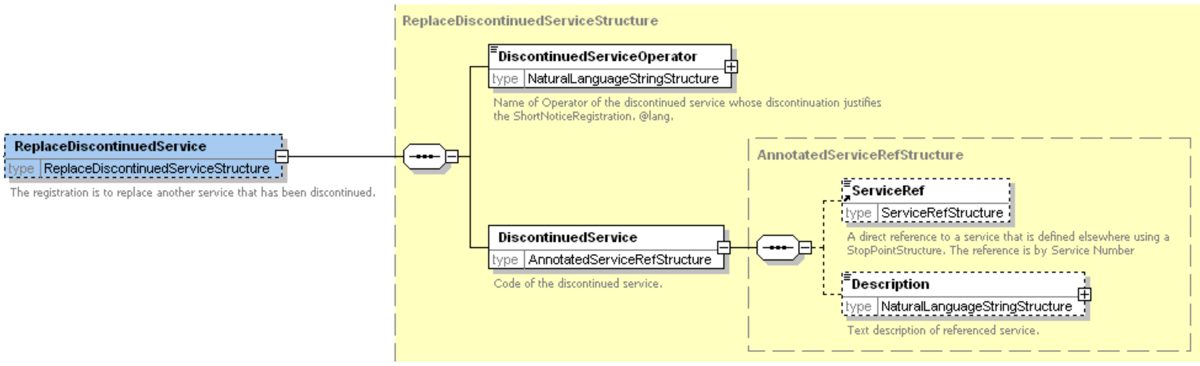


Figure 6-48 – ShortNoticeRegistration / ReplaceDiscontinuedService Element

6.5.13.5 ShortNoticeRegistration / LocalHolidayChange Element

The ***LocalHolidayChange*** element (*Figure 6-49*) identifies the local holiday which justifies the short notice registration.

- ***LocalHolidayNote***: Description of local holiday.

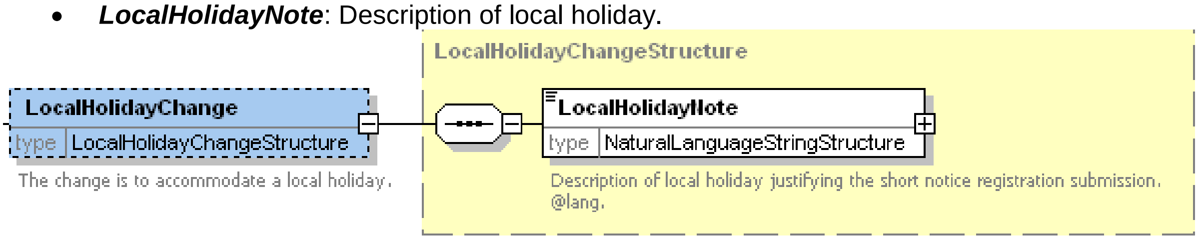


Figure 6-49 – ShortNoticeRegistration / LocalHolidayChange Element

6.5.13.6 ShortNoticeRegistration / SpecialOccasion Element

The ***SpecialOccasion*** element (*Figure 6-50*) identifies the special occasion which justifies the short notice registration.

- ***SpecialOccasionName***: Name of special occasion.

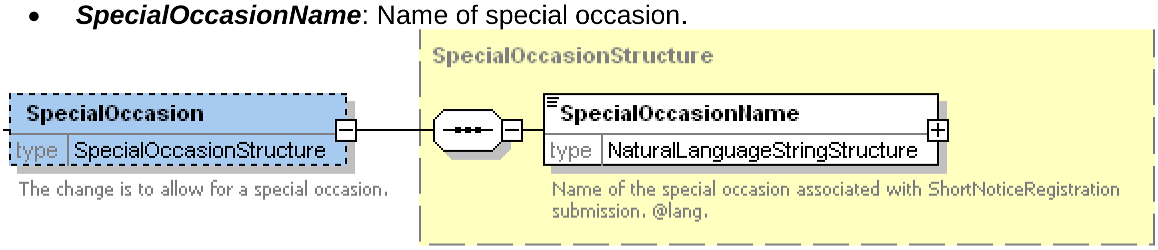


Figure 6-50 – ShortNoticeRegistration / SpecialOccasion Element

6.5.13.7 ShortNoticeRegistration / RegulationOrderCompliance Element

The ***RegulationOrderCompliance*** element (*Figure 6-51*) identifies whether the short notice registration is to comply with a regulation order.

- ***TrafficOrderNote:*** Identifies the order.

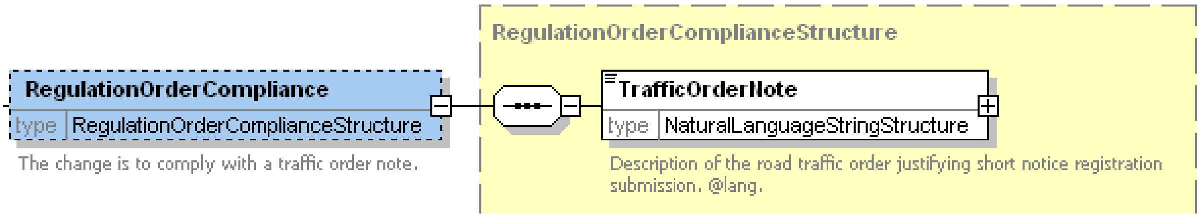


Figure 6-51 – ShortNoticeRegistration / RegulationOrderCompliance Element

6.5.13.8 ShortNoticeRegistration / ChangeRequestedByExternalAuthority Element

The ***ChangeRequestedByExternalAuthority*** (*Figure 6-52*) specifies whether the short notice registration is needed to meet a request by an external authority such as the Police, and any explanation or corroboration of the change.

- ***ChangeRequestDescription***: Explanation or corroboration of why the change is required.

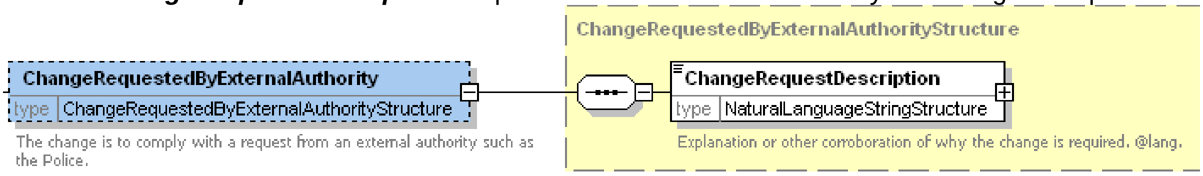


Figure 6-52 – ShortNoticeRegistration / ChangeRequestedByExternalAuthority Element

6.5.13.9 ShortNoticeRegistration / ExceptionalRequirement Element

The ***ExceptionalRequirement*** element (*Figure 6-53*) specified whether the registration is needed to meet an allowed exceptional requirement.

- ***ChangeRequestDescription***: Explanation or corroboration of why the change is required.

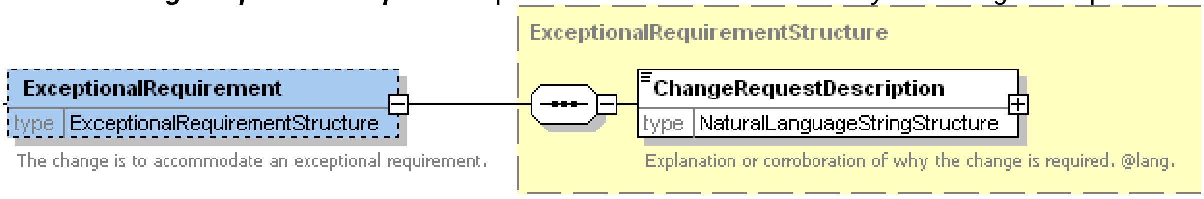


Figure 6-53 – ShortNoticeRegistration / ExceptionalRequirement Element

6.6 Service Description Elements

6.6.1 Services Element

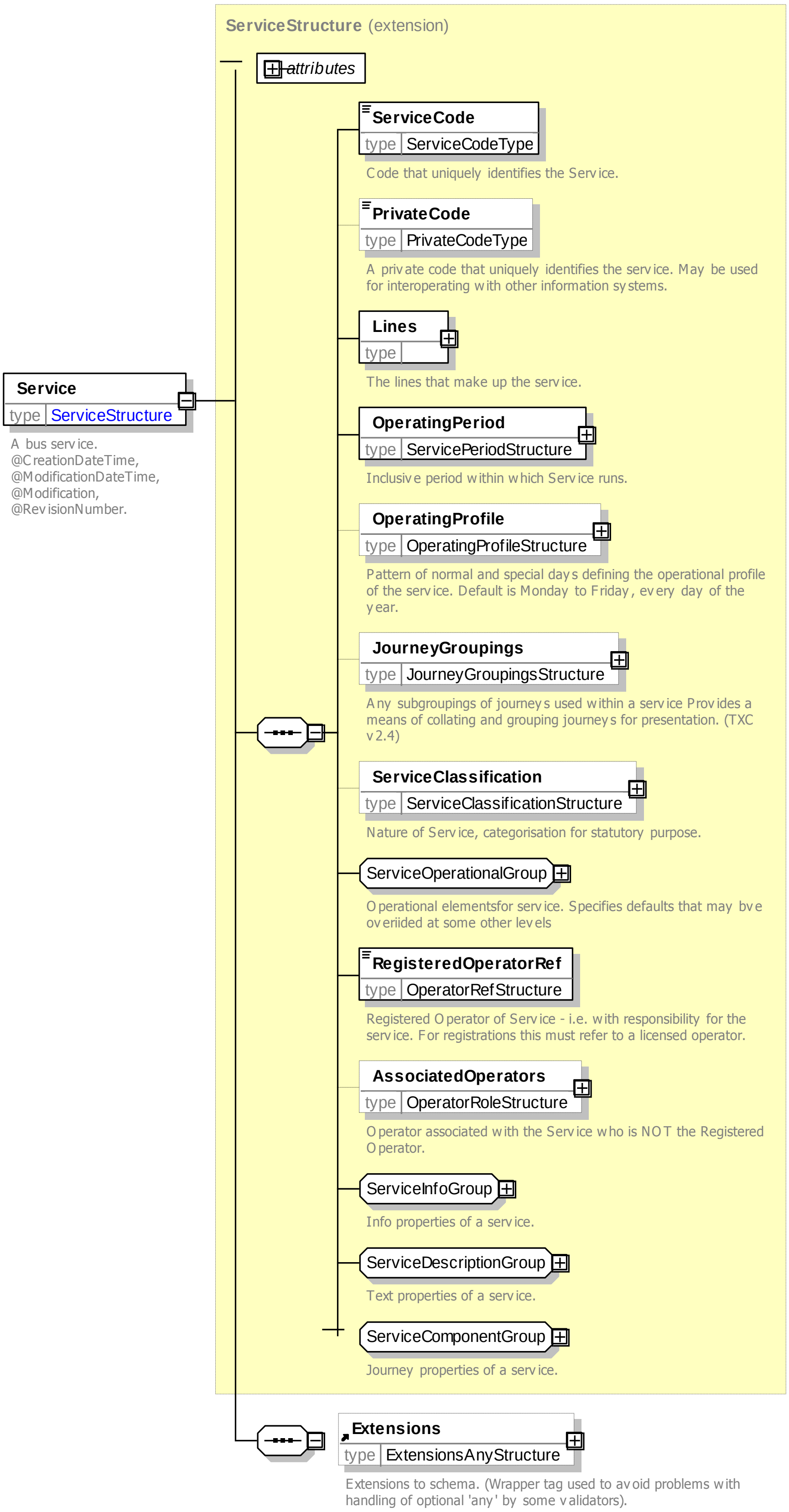
Definitions of each ***Service*** describing a bus schedule are contained within the ***Services*** container element***:***

- In a *TransXChange Registration* schema document, only one registered service may be described at a time. The registered ***Service*** must reference a ***Registration***, and the referenced ***Registration*** must describe the operator in full with a ***LicensedOperator***. Instance. Relevant details of other connecting services may be included in the document as separate service declarations.
- In a *TransXChange General* Schema document, many services can be described.

6.6.2 Service Element

The ***Service*** element (*Figure 6-54*) describes a service. The elements include:

- ***ServiceCode***: The unique identifier for the ***Service***. This is of type string (+TXC v 2.4) to allow arbitrary characters to be used. Note that is using an Ampersand ('&') the '&' will have to be escaped, for example *'S2&S2'* must be coded <***ServiceCode***>*'S1&S2'*</***ServiceCode***>
- ***PrivateCode***: An identifier for the ***Service*** that can be used to associate it with other systems.
- ***Lines***: The public identifiers for the ***Service***. See later.
- ***OperatingPeriod***: Period within which ***Service*** operates. See below.
- ***OperatingProfile:*** Default operational days for journeys running the ***Service***. See Operational Days elements later.
- ***JourneyGroupings:*** Journey Groupings for the ***Service***. See Below.
- ***ServiceClassification***: Type of the ***Service***. See below.
- ***ServiceOperationalGroup:*** Default operational elements associated with the ***Service***. See below.
- ***RegisteredOperatorRef***: Registered operator of the ***Service***. See ***LicensedOperator*** and ***Operator***. On a ***Registration Service*** this must reference a ***LicensedOperator*** instance.
- ***AssociatedOperatorRef:*** Another operator associated with the service in a secondary capacity. See ***Operator*** and ***LicensedOperator***.
- ***ServiceInfoGroup:*** Further informational elements about the ***Service***. See below.
- ***ServiceDescriptionGroup:*** Further descriptive elements about the service. See below.
- ***ServiceComponentGroup:*** Information about the routes and journeys patterns comprising the ***Service***. See below.



•

Figure 6-54 – Service Element

6.6.3 Service / ServiceInfoGroup

The ***ServiceInfoGroup*** (*Figure 6-55*) group holds informational elements describing the ***Service***.

- ***ServiceHasMirror***: Whether service has a corresponding service in the return direction.
- ***StopRequirements:*** Whether the service requires new stop declarations. See below.
- ***Mode***: Transport mode of service. See *Table 6-21*. Default is bus. The equivalent NeTEx modes are shown in the right-hand column.

Value	Description	NeTEx
air	Air service.	*air*
bus	Bus service.	*bus*
coach	Coach service.	*coach*
ferry	Ferry service.	*water **
metro	Metro service.	*metro*
telecabine	Lift or cableway service	*cableway **
train	Train service.	*train*
tram	Tram service.	*tram*
underground	Underground service.	*metro*

Table 6-21 – Allowed Values for Service / Mode

- ***PublicUse***: Whether service allows public use, i.e. is not 'Closed Door'.
- ***ServiceAvailability***: Whether service has a corresponding service in the return direction. See below.
- ***Express***: Whether service is flagged as an express (i.e. limited stop) service.
- ***CommercialBasis:*** on which the service is offered. May be overridden for specific journey patterns, vehicle Journeys and timing links. See Table 6-22 for allowed values. (+TXC v2.4)

Value	Description
contracted	Service on Link is contracted
notContracted	Service on Link is not contracted.
partContacted	Service on Link is part contracted.
unknown	Basis is unknown.

Table 6-22 – Allowed Values for Service / CommercialBasis

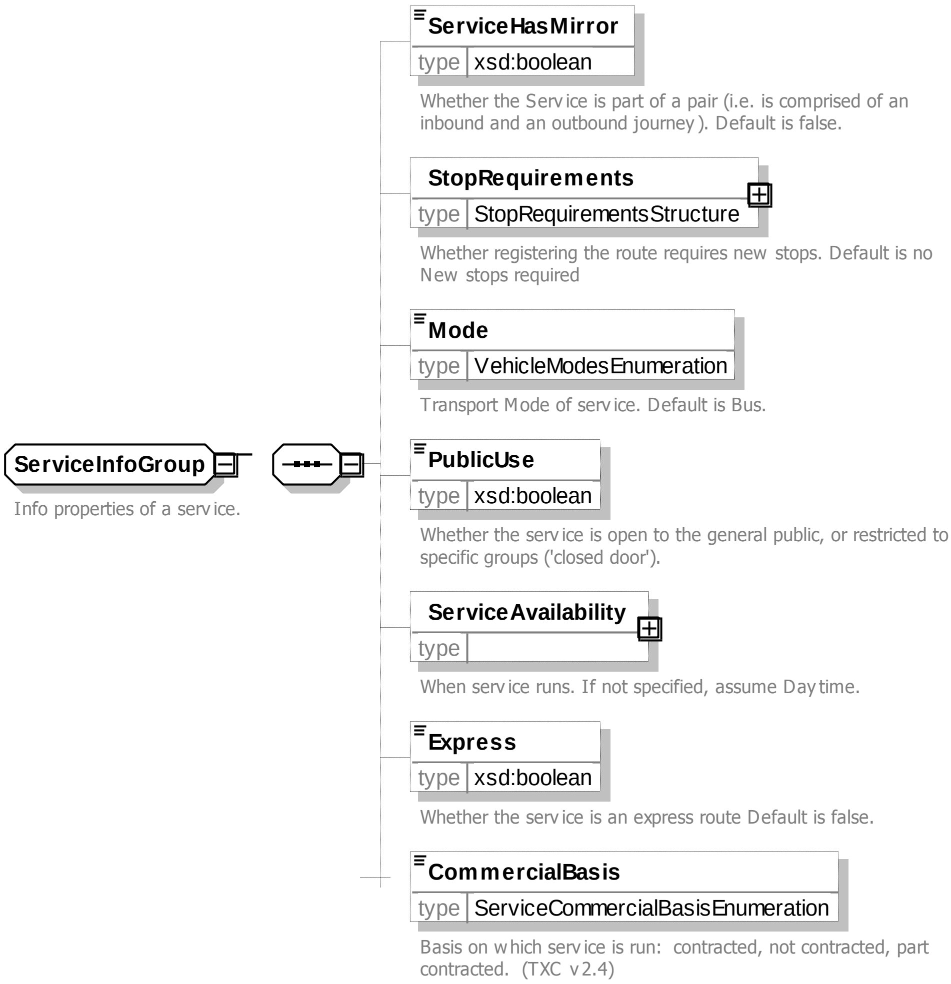


Figure 6-55 – Service / ServiceInfoGroup

6.6.4 Service / ServiceDescriptionGroup

The ***ServiceDescriptionGroup*** (*Figure 6-56***)** group holds ancillary descriptive elements describing the Service.

- ***Description:*** Text description of the services. On registrations should include “A description of the service or change for Notices & Proceedings”. For example, “*a regular service at half-hourly intervals daytime on Mondays to Saturdays, and hourly in the evenings and on Sundays*".
- ***Note:*** Structured notes associated with service. See common schema elements later.
- ***SchematicMap:*** Name of any schematic map associated with services. File name. Must be an image file (png, gif, jpeg). Schematic maps must be provided for Registrations.
- ***MarketingName***: Name to use when displaying service in some applications. (+TXC v2.4)

ToBeMarketedWith: Information on marketing of the services. See below.

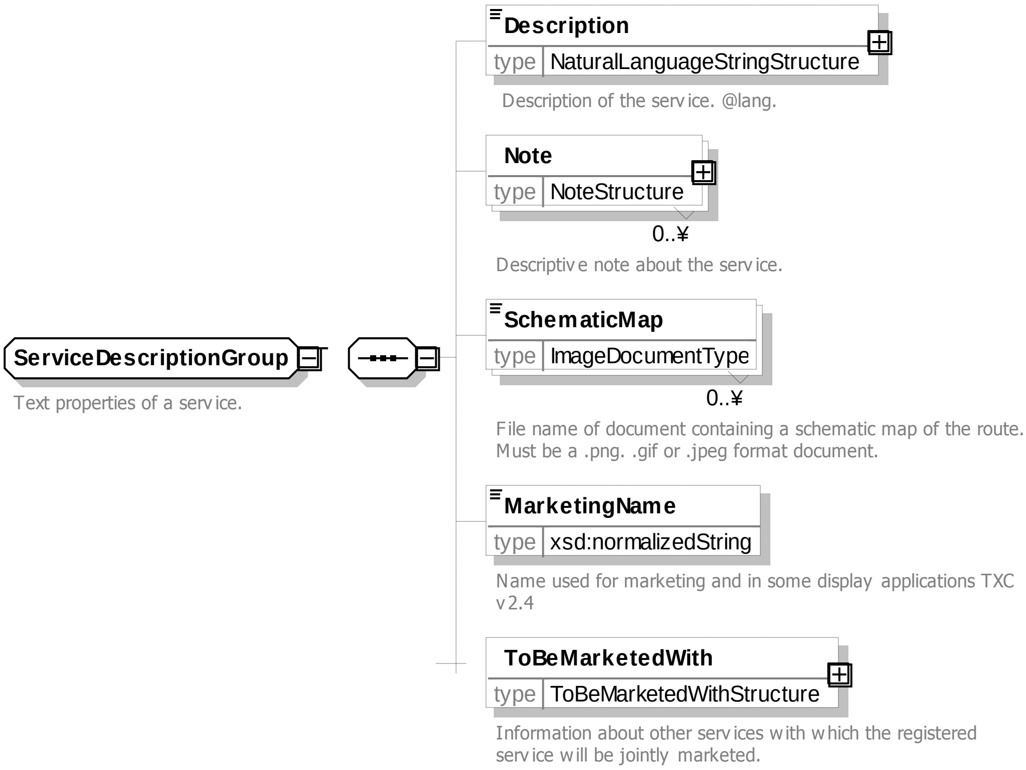


Figure 6-56 – Service / ServiceDescriptionGroup

6.6.5 Service / ServiceComponentGroup

The ***ServiceComponentGroup*** (*Figure 6-57***)** holds the fundamental timetable components of the Service.

- ***StandardService:*** Any standard service component.
- ***FlexibleService:*** Any flexible service component.
- ***Direction:*** The direction of the ***Service***. See *Table 6-2*

Value	Description
inbound	Inbound Direction.
outbound	Outbound Direction.
inboundAndOutbound	Inbound and Outbound Direction.
circular	Circular Direction.
clockwise	Clockwise Direction.
antiClockwise	Anti-Clockwise Direction.

Table 6-23 – Allowed Values for Service / Direction

- ***JourneyPatternInterchange***: Zero or more interchanges at which the journey patterns of the service connect.

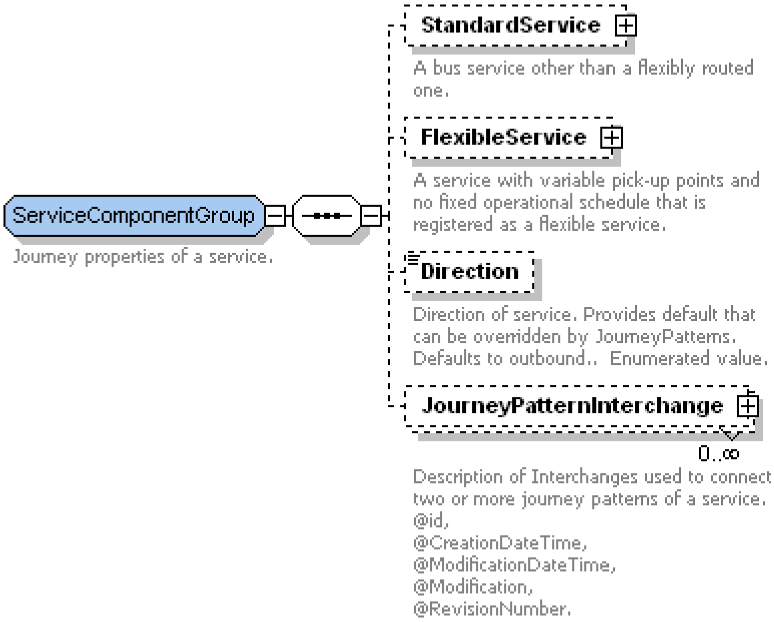


Figure 6-57 – Service / ServiceComponentGroup

6.6.6 Service / Subelements

6.6.6.1 Service / Line Element

The ***Line*** element (*Figure 6-58*) allows one or more public identifiers of the service to be associated with the vehicle journeys of the service. For example, lines ‘1’, ‘1a’, ‘1b’. Each individual ***VehicleJourney*** element specifies the line or line variant that the journey runs. A ***Line*** provides an arbitrary label for presentational and marketing purposes and does not necessarily correspond to the strict route variants: the same line name may be used on services with different stopping patterns. A ***Line*** is identified by a unique ***id*** attribute.
Each ***Line*** has:

- ***LineName***. Name of line, typically a number or letters & number
- ***MarketingName***: Alternative marketing name for Line (+TXC v2.4).
- ***Outbound Description***: Description of outbound direction of line for publicity purposes. (+TXC v2.4). See below
- ***Inbound Description***: Description of inbound direction of line for publicity purposes. (+TXC v2.4). See below
- ***LineFontColourGroup***: (+TXC v2.4).Optional preferred colour to use for when showing line labels in visual media. See below.
- ***LineImage***: Url of an optional image associated with line to use in graphic media. (+TXC v2.4)

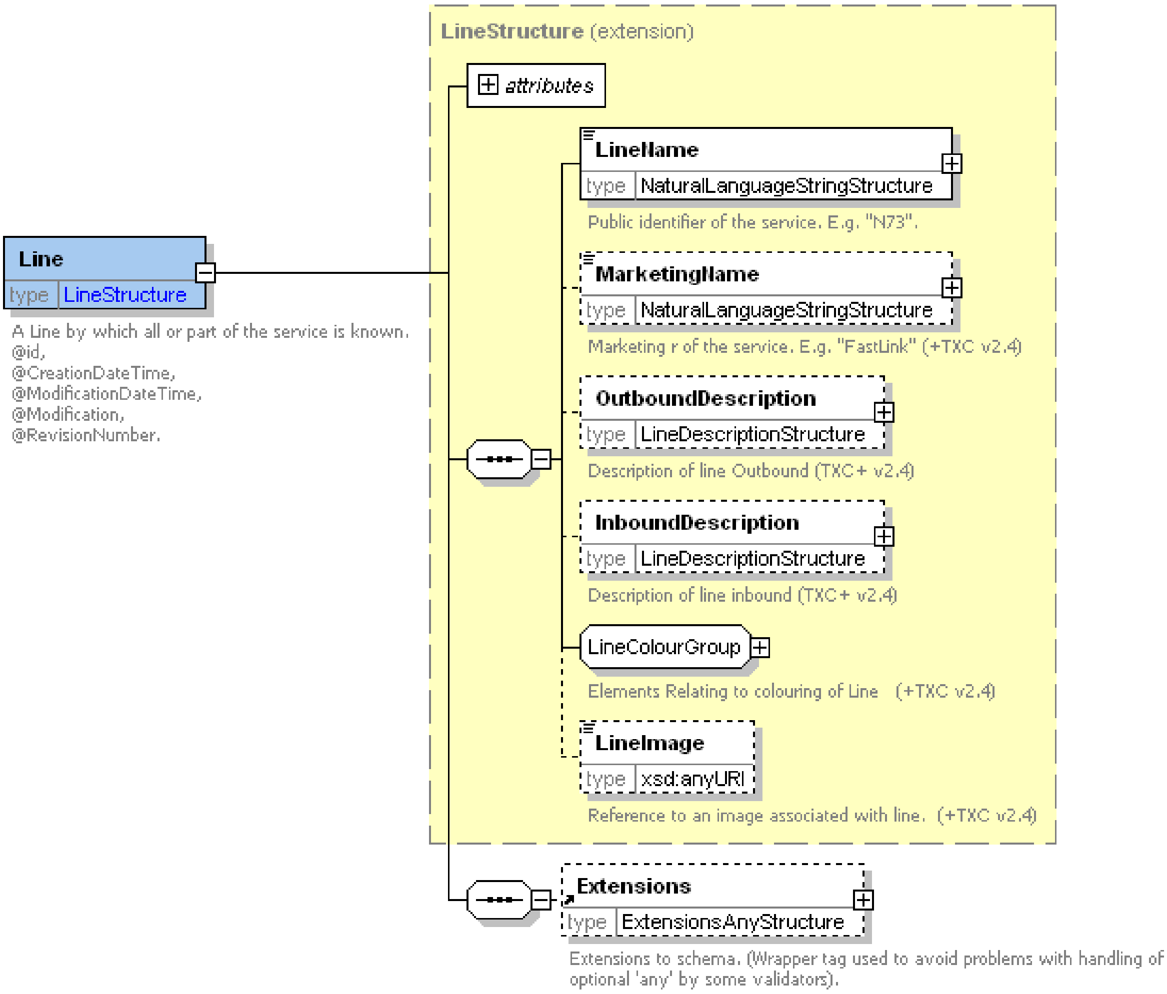


Figure 6-58 – Service / Line Element

6.6.6.2 Line / Description Element (+TXC v2.4)

The ***LineDescription*** element (*Figure 6-58*) describes a direction of a line for publicity purposes

- ***Origin***. Origin of line. (+TXC v2.4).
- ***Destination***. Destination of line. (+TXC v2.4).
- ***Vias***. Via points. (+TXC v2.4).
  - ***Via***. Via point on line. (+TXC v2.4).
- ***Description***. Description of line. (+TXC v2.4).

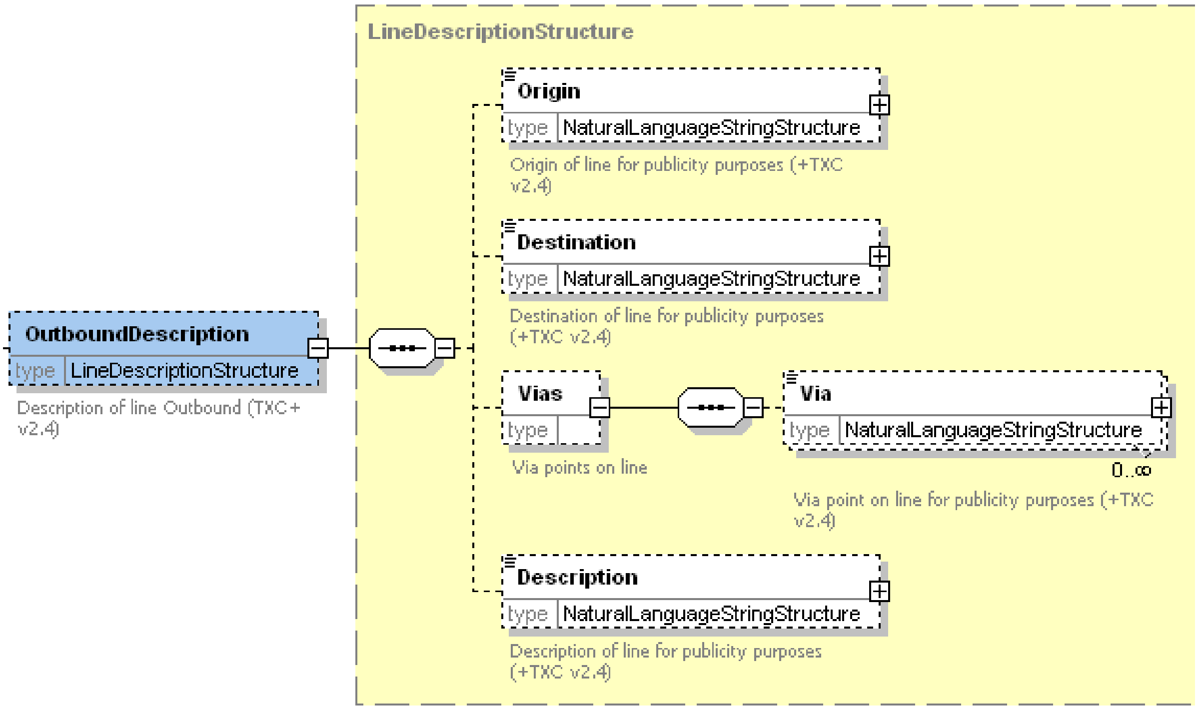


Figure 6-59 – Service / Line Element

6.6.6.3 LineColour Group (+TXC v2.4)

The ***LineColourGroup*** (*Figure 6-58*) specifies colour preferences for presenting the line consistently in media.

- ***LineColour***: Optional preferred colour to use when showing line in graphic media. (+TXC v2.4)
- ***LineFontColour***: Optional preferred colour to use for text when showing line labels in visual media. (+TXC v2.4)
- ***AlternativeLineColour***: Second choice colour to use when showing line in graphic media. (+TXC v2.4)
- ***LineColour***: Second choice colour to use when showing Line labels in visual media. (+TXC v2.4)

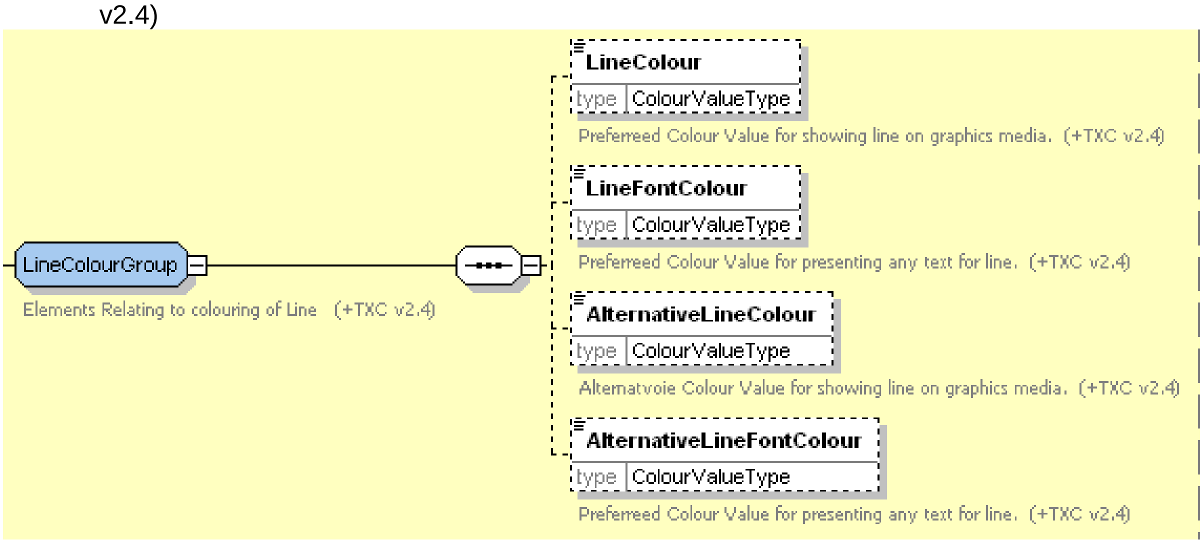


Figure 6-60 – Line / LineColourGroup

6.6.6.4 Service / OperatingPeriod Element

The ***OperatingPeriod*** element (*Figure 6-61*) states the period over which the ***Service*** operates. It includes

- A ***StartDate***: Date at which service commences.
- An ***EndDate***. Date at which service ends. If absent, continues indefinitely.
- Recommended EndDate: Even if service is notionally open ended for registration purposes, there may be a recommended date after which not to use the data. (TXC v2.4)

See also ***OperationProfile*** element for further elements relating to the operating days of a service.

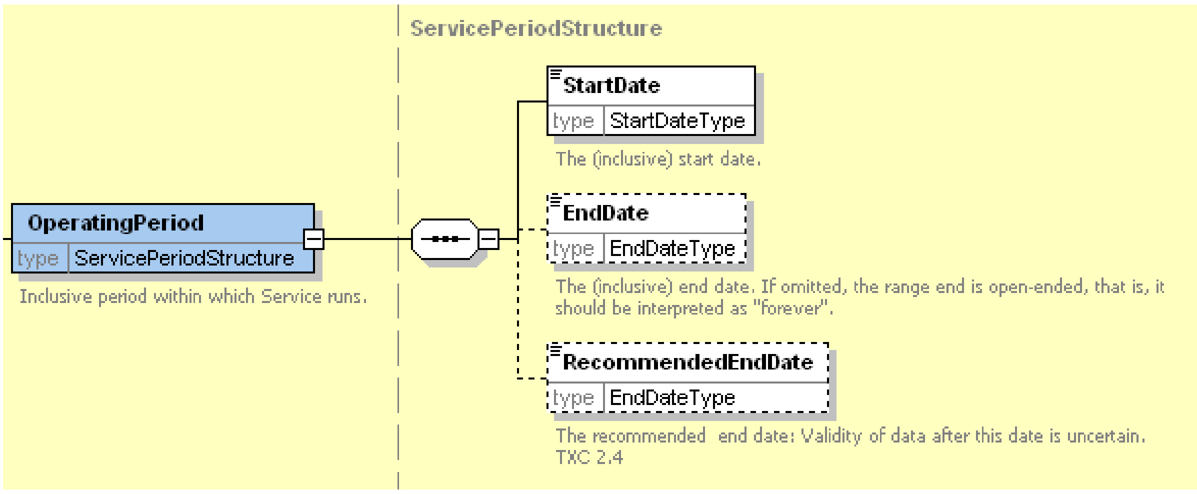


Figure 6-61 – Service / OperatingPeriod Element

6.6.6.5 Service / ServiceClassification Element

The ***ServiceClassification*** element (*Figure 6-62*) classifies the service as being one or more of a number of categories of service. The classifications are as follows:

- ***NormalStopping:*** A service where all stops on a route are used.
- ***LimitedStops:*** A service where only certain pre-defined stops on a route are used.
- ***HailAndRide:*** A service that stops anywhere on designated parts of the route, if flagged down by passengers where it is safe to do so.
- ***Flexible:*** A service running in accordance with the rules for a flexible service, with designated pickup and set down zones or points. Must be specified if service is a ***FlexibleService***.
- ***ExcursionOrTour:*** A service where all passengers go to the same destination and return to their departure point. Further qualified by:
    - ***MaxDepartures***: Maximum number of vehicle departures within one day associated with an excursion type service.
- ***RuralService:*** A service primarily aimed at serving rural communities (i.e. at locations with populations less than 25,000 people).
- ***SchoolOrWorks:*** A service dedicated to a school or works that is not available to the public.
- ***OtherService:*** Services that do not fit any of the defined categories. Should only be used sparingly:
    - Further explained by a ***Description***.

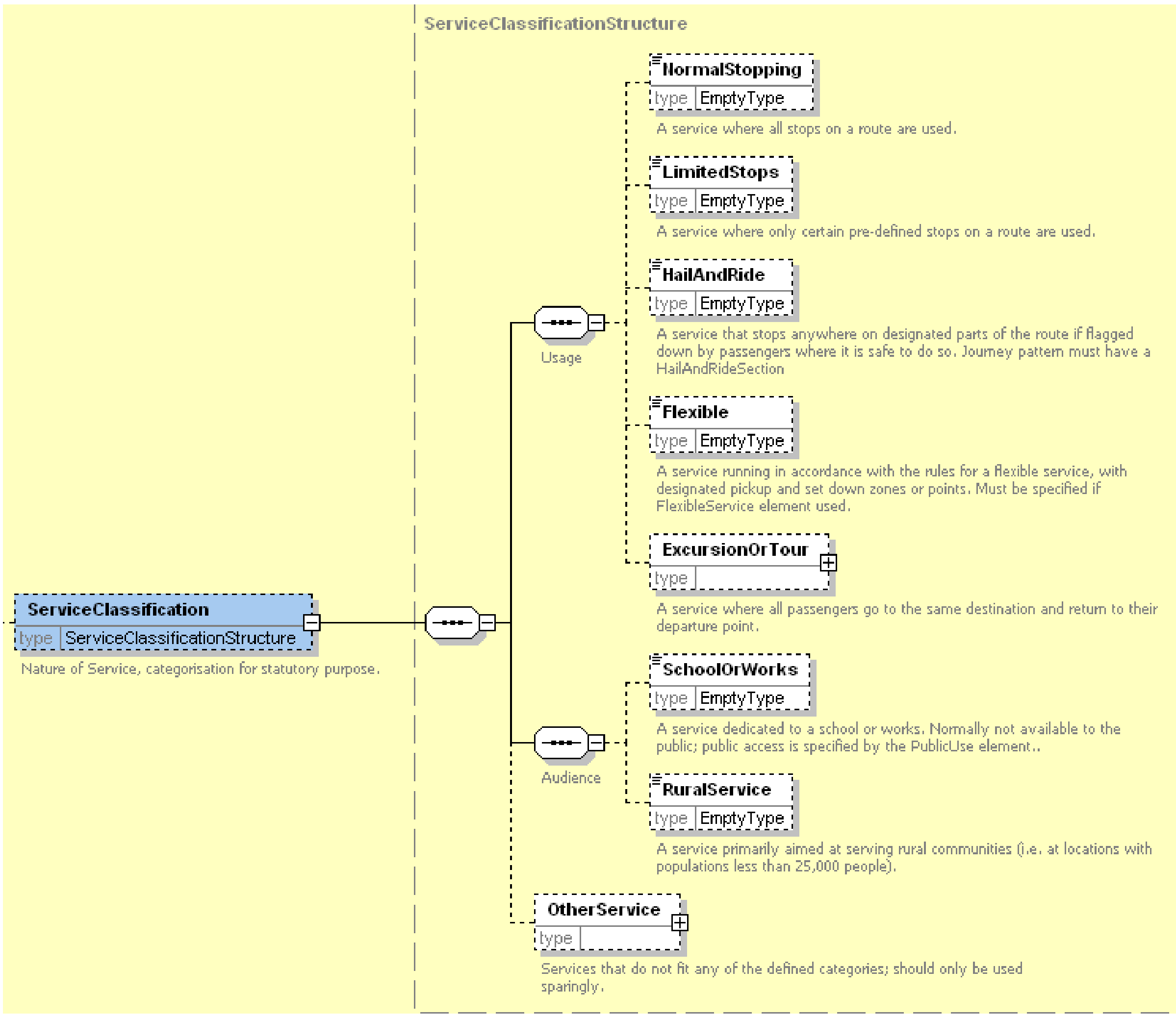


Figure 6-62 – Service / ServiceClassification Element

Service types may be combined in any way. Normal combinations of service are shown in *Table 6-24*:

Group	ServiceClassification	Normal Stop-ping	Limited Stops	Hail And Ride	Flexible Service	Excursion Or Tour	Other Service	School Or Works	Rural Serv-ice
Operation	***NormalStopping***	-	N	Y	Y	Y	Y	Y	Y
	LimitedStops	N	-	N	N	Y	Y	Y	Y
	HailAndRide	Y	N	-	N	N	N	Y	Y
	FlexibleService	Y	N	N	-	N	N	Y	Y
	ExcursionOrTour	N	N	N	N	-	N	Y	Y
	OtherService	Y	Y	N	N	N		Y	Y
Purpose	***SchoolOrWorks***	Y	Y	Y	Y	N	N	-	Y
	RuralService	Y	Y	Y	Y	N	N	Y	-

Table 6-24 – Allowed ServiceClassification Combinations

6.6.6.6 Service / ServiceOperational Group

The ***ServiceOperationalGroup*** (Figure 6-56**)** group holds ancillary descriptive elements describing operational data associated with the ***Service***.

- ***TicketMachineServiceCode:*** Unique Identifier associated with s ***Service*** for use in ticketing machine systems. May be overridden on Individual Journey Patterns & Vehicle Journey instances.
- ***VehicleType***: The type of vehicle normally used on the service – can be used to specify if the ***Service*** is normally considered to be wheelchair accessible. More specific details may be

provided by the ***VehicleType*** / ***Wheelchair VehicleEquipment***. (+TXC v2.4) See later below.

- ***AssistanceService***: The assistance normally available on the service – can be overridden on individual journeys (+TXC v2.5). See later below.
- ***ServiceFacilitySet***: The facilities normally available on the service – can be overridden on individual journeys (+TXC v2.5). See later below.

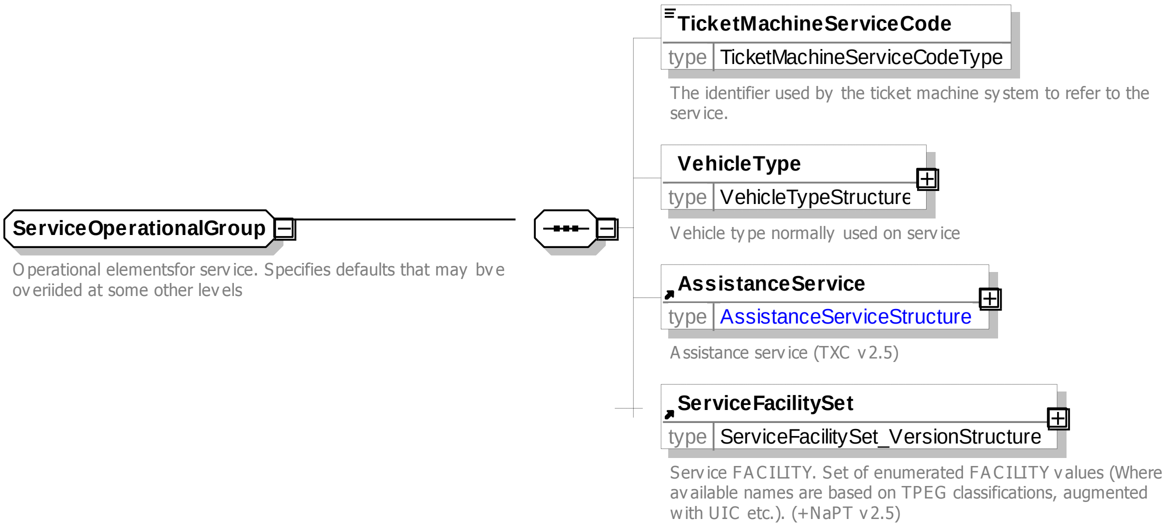


Figure 6-63 – Service / ServiceOperationalGroup

6.6.6.7 ServiceOperationalGroup / AssistanceService Element (+TXC V2.5)

The ***AssistanceService*** element *(*Figure 6-64) describes the available assistance for wheelchair users to access a vehicle. (+TXC V2.5)

- ***Name:*** Description of service). (+TXC v2.5.)
- ***Availability:*** Days or times when service is available. (+TXC v2.5.) If not specified assume always available.
  - **DayType:** Day type and Timeband when service is available.
- ***AssistanceServiceType:*** Assistance services for wheelchair users. See Table 6-25. (+TXC v2.5.)

• Value	Description
none	No Assistance service is available
boardingAssistance	Assistance is available to board the vehicle.
wheelchairAssistance	Assistance is available for wheelchair users
conductor	A conductor is available
unknown	Not known if available.

Table 6-25 – Allowed Values for AssistanceServiceType (+TXC v2.5)

- ***AssistanceAvailability:*** Availability of services for wheelchair users. Table 6-8 earlier shows the allowed values for assistance availability. (+TXC v2.5.)
- ***Staffing***: shows the normal staffing arrangements for the stop. See **Table 6-26** for allowed values

Value	Description
fullTime	Staffed full time during opening hours
partTime	Only staffed at certain times
unmanned	Not normally staffed
unknown	Staffing unknown

Table 6-26 – Allowed Values for Staffing (+TXC v2.5)

- ***AccessibilityTools***: Devices available for use by passengers. One or more - See Table 6-27 for allowed values.

Value	Description
wheelchair	Wheelchairs are available for use.

walkingStick	Walking sticks are available for use.
audioNavigator	Audio Navigators are available for use.
visualNavigator	Visual Navigator aids are available for use, e.g. floor strips.
passengerCart	Passenger carts are available for use.
other	Other equipment is available for use.

Table 6-27 – Allowed Values for AccessibilityTools (+TXC v2.5)

- ***Languages***: Languages spoken for assistance. One or more languages, as specified by XML Language value. (e.g. en, cy, de, fr, etc). (+TXC v2.5.)
- ***AccessibilityTrainedStaff***: Staff is trained to assist wheelchair users. (+TXC v2.5.)
- ***EmergencyServices:*** List of available emergency service support. One or more - See Table 6-28 for allowed values. (+TXC v2.5.)

Value	Description
police	Police are available at location or on board.
firstAid	First aid is available at location or on-board.
sosPoint	SOS help Points are available at location or on board.
ccTv	CCTV covers location or on board.
other	Other services available at location or on board.

Table 6-28 – Allowed Values for EmergencyServices (+TXC v2.5.)

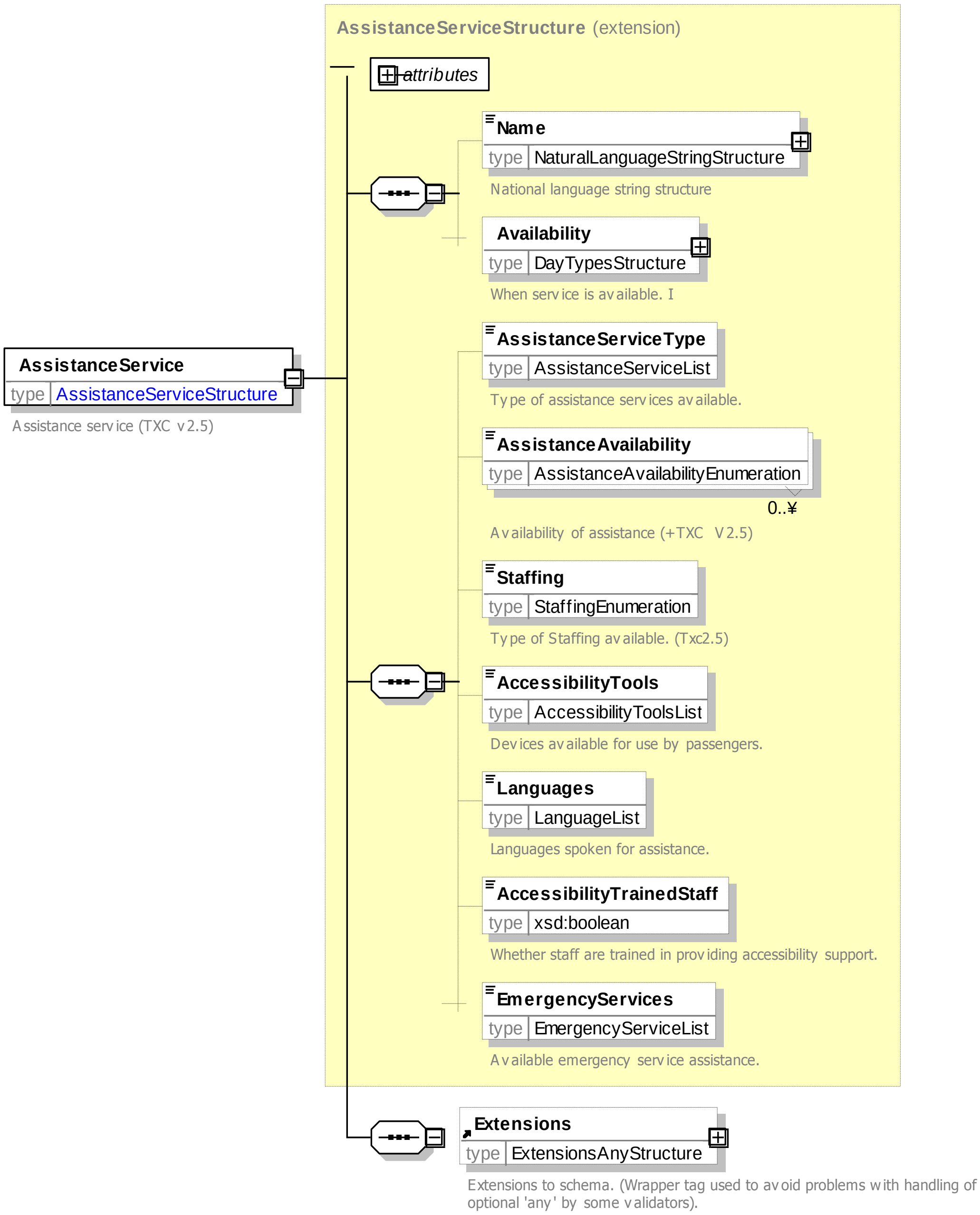


Figure 6-64 – AssistanceService Element (+TXC v2.5)

6.6.6.8 ServiceOperationalGroup / ServiceFacilitySet Element (+TXC V2.5)

The ***ServiceFacilitySet*** element (Figure 6-65) describes the available on-board services for a vehicle as simple lists of named facilities. Certain services with complex properties can be additionally described by an equipment element, for example ***SanitaryFacilityEquipment*** allows detailed properties of ***SanitaryEquipment*** to be specified. (+TXC V2.5).

- ***Name:*** Description of service). (+TXC v2.5.)

CommonFacilityGroup describes services that may also apply to a Site:

- ***CateringFacilityList:*** Catering users. See **Table 6-29**. (+TXC v2.5.)

• **Value**	**Description**
none	No Assistance service is available
boardingAssistance	Assistance is available to board the vehicle.
wheelchairAssistance	Assistance is available for wheelchair users
conductor	A conductor is available
unknown	Not known if available.

Table 6-29 – Allowed Values for *CateringFacilityList*

- ***FareClasses:*** (+TXC v2.5) specifies the classes of fare available. See Table 6-30 for allowed values.

Value	Description
firstClass	First class seats
secondClass	Second or standard class seats
thirdClass	Third class seat
businessClass	Business class
economyClass	Economy Class
unknown	Not known if available.

Table 6-30 – Allowed values for ServiceFacilitySet / FareClasses (+TXC v2.4)

- ***NuisanceFacilityList:*** (+TXC v2.5) The type of accommodation available. See Table 6-31 for allowed values.

Value	Description
smoking	Smoking zone.
noSmoking	No smoking. zone.
mobilePhoneUseZone	Mobile phone zone.
mobilePhoneFreeZone	Quiet area.
famillyArea	Family area
childfreeArea	Child free area.
other	Other area.
smoking	Smoking zone.
noSmoking	No Smoking. zone.
lmobilePhoneUseZone	Mobile phone zone.
mobilePhoneFreeZone	Quiet area.
famillyArea	Family area

Table 6-31 – Allowed values for ServiceFacilitySet / NuisanceFacility (+TXC v2.5)

- ***PassengerCommsFacilityList:*** (+TXC v2.5) The types of on-board passenger communications and entertainment facilities. See Table 6-32 for allowed values.

Value	Description
freeWifi	Free wifi.
publicWifi	Public wifi, may be charged.
telephone	Telephone.
internet	Internet access.
mobileCoverage	Mobile phone coverage.
videoEntertainment	Video entertainment.
audioEntertainment	Audio entertainment.
powerSupplySockets	Power supply sockets.

Table 6-32 – Allowed values for ServiceFacilitySet / PassengerCommsFacilityList (+TXC v2.5)

- ***PassengerInformationFacilityList:*** (+TXC v2.5) The type of passenger information facilities available on-board. See Table 6-33 for allowed values.

Value	Description
audioInformation	Audio information.
audioForHearingImpaired	Special Audio information for those who are hearing impaired.
visualDisplays	Visual displays of information.
displaysForVisuallyImpaired	Specially enhanced visual information for those who are visually impaired.
largePrintTimetables	Large print timetables.
other	Other INFO FACILITY.

Table 6-33 – Allowed values for ServiceFacilitySet / PassengerInformationFacility (+TXC v2.5)

- ***SafetyFacilityList:*** (+TXC v2.5) Some on-board capabilities that enhance to passenger safety. See **Table 6-34** for allowed values.

Value	Description
mobileCoverage	There is mobile phone coverage.
ccTv	There is close circuit television camera surveillance.
sosPoint	There is an SOS point.
staffed	There are staff.

Table 6-34 – Allowed values for ServiceFacilitySet / SafetyFacility (+TXC v2.5)

- ***SanitaryFacilityList:*** (+TXC v2.5) The types of lavatory and washing facilities available on service. See **Table 6-35** for allowed values.

Value	Description
toilet	Lavatories.
wheelchairAccessToilet	Toilet for wheelchair access.
shower	Shower.
wheelchairBabyChange	Baby change facility for wheelchair users.
babyChange	Baby change facility.
washingAndChangeFacilities	Wash and Change facilities.
other	Other sanitary facility.

Table 6-35 – Allowed values for ServiceFacilitySet / SanitaryFacility (+TXC v2.5)

ServiceFacilityGroup describes services specific to a Service:

- ***AccommodationFacilityList:*** (+TXC v2.5) The type of accommodation available. See **Table 6-36** for allowed values.

Value	Description
seating	Seating.
sleeper	Sleeper accommodation.
doubleSleeper	Double Sleeper accommodation.
singleSleeper	Single Sleeper accommodation.
specialSleeper	Special Sleeper accommodation.
couchette	Couchette.
recliningSeat	Reclining seat.
babyCompartment	Baby Compartment.
familyCompartment	Family Compartment.
panoramaCoach	Panorama coach.
pullmanCoach	Pullman coach.
standing	Standing accommodation.

Table 6-36 – Allowed values for ServiceFacilitySet / AccommodationFacility (+TXC v2.5)

- ***LuggageCarriageFacilityList:*** (+TXC v2.5) describes the type of l available. See **Table 6-37** for allowed values.

Value	Description
noBaggageStorage	No baggage storage.
luggageRacks	Luggage racks.
extraLargeLuggageRacks	Extra-large luggage storage.
baggageVan	Baggage van.
cyclesAllowed	Cycles allowed.
cyclesAllowedInVan	Cycles allowed in van.
cyclesAllowedInCarriage	Cycles allowed in carriage.
cyclesAllowedWithReservation	Cycles allowed with reservation.
noCycles	No cycles allowed.
noBaggageStorage	No baggage storage.
luggageRacks	Luggage racks.
extraLargeLuggageRacks	Extra-large luggage storage.

Table 6-37 – Allowed values for ServiceFacilitySet / LuggageCarriage (+TXC v2.5)

- ***ReservationFacilityList:*** (+TXC v2.5) Date that ***Service*** starts.

Value	Description
seatReservationsCompulsory	Seat Reservations Compulsory
cycleReservationsCompulsory	Bicycle Reservations Compulsory
seatReservationsRecommended	Seat Reservations Recommended
seatReservationsPossible	Seat Reservations Possible
wheelchairOnlyReservations	Reservations are required for Wheelchair but not necessarily for others

Table 6-38 – Allowed values for ServiceFacilitySet / Reservations (+TXC v2.4)

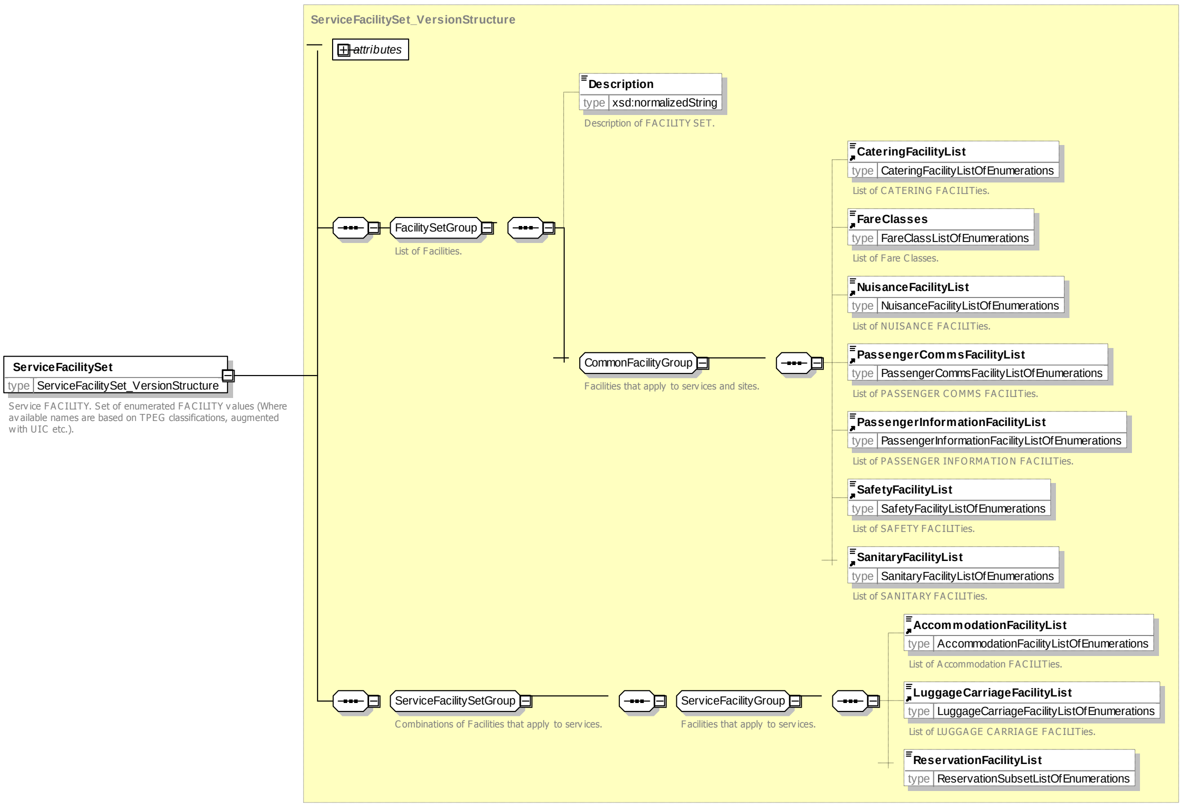


Figure 6-65 – ServiceFacilitySet Element (+TXC v2.5)

6.6.6.9 Service / AssociatedOperators Element

The ***AssociatedOperators*** (*Figure 6-66*) element records details about any operators associated with the service other than the registered operator. The ***AssociatedOperator*** comprises:

- ***OperatorRef:*** Reference to an ***Operator*** or ***LicensedOperator*** definition. See above.
- ***Role:*** Description of the role of the associated operator.

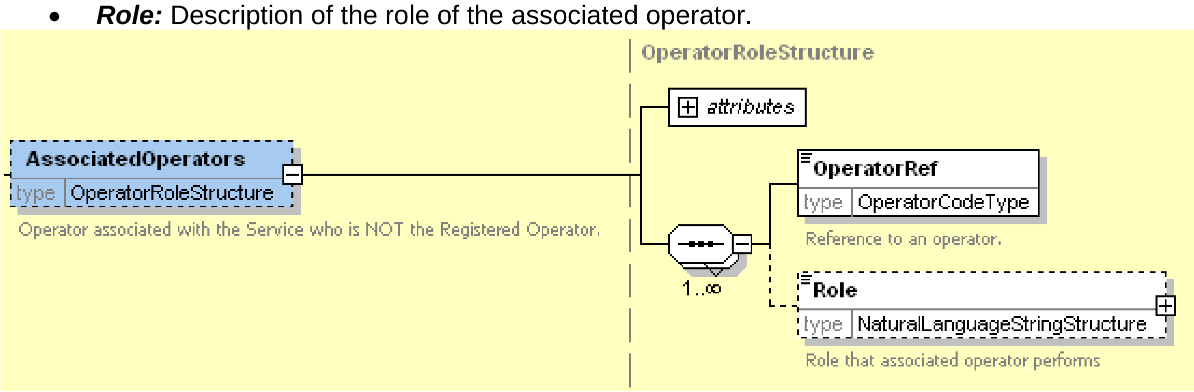


Figure 6-66 – Service / AssociatedOperators Element

6.6.6.10 Service / StopRequirements Element

The ***StopRequirements*** element (*Figure 6-67*) specifies whether a service does or does not require any new stops, and provides a list of these stops. If not, specifies, default is to assume ***NoNewStopsRequired***. ***StopRequirements*** does not have to be specified on a Cancellation.

- ***NoNewStopsRequired:*** No new stops are needed.
- ***NewStops:*** New stops are needed.
  - ***StopPointRef***: Reference to identify the new stop.
  - ***Note:*** Optional explanatory note accompanying definition.

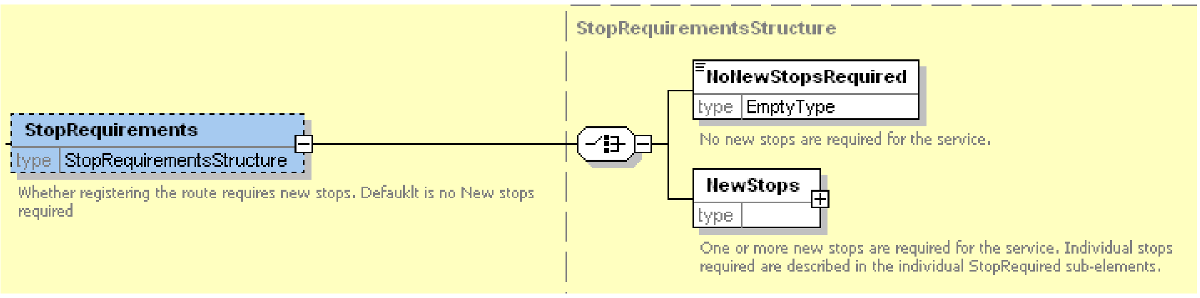


Figure 6-67 – Service / StopRequirements Element

Note that an explicit list of the new stops associated with the service is useful to the EBSR registration authorities. That a stop is relevance to a particular service cannot be inferred merely from the fact that a new stop has been added to NaPTAN. The NewStops tag should list the identifiers of the new stops. Details of the stops may already have been added to NaPTAN or may be declared locally.

6.6.6.11 Service / ServiceAvailability Element

The ***ServiceAvailability*** element (*Figure 6-68*) specifies the time of day a service runs TIME DEMAND TYPE) as a broad classification. One of the following:

- ***TwentyFourHours:*** Service runs all day and all night continuously.
- ***Daytime:*** Service runs in daytime.
- ***Peak:*** Service runs in peak hours only.
- ***OffPeak:*** Service runs in off-peak hours only.
- ***Night:*** Service is a night service.

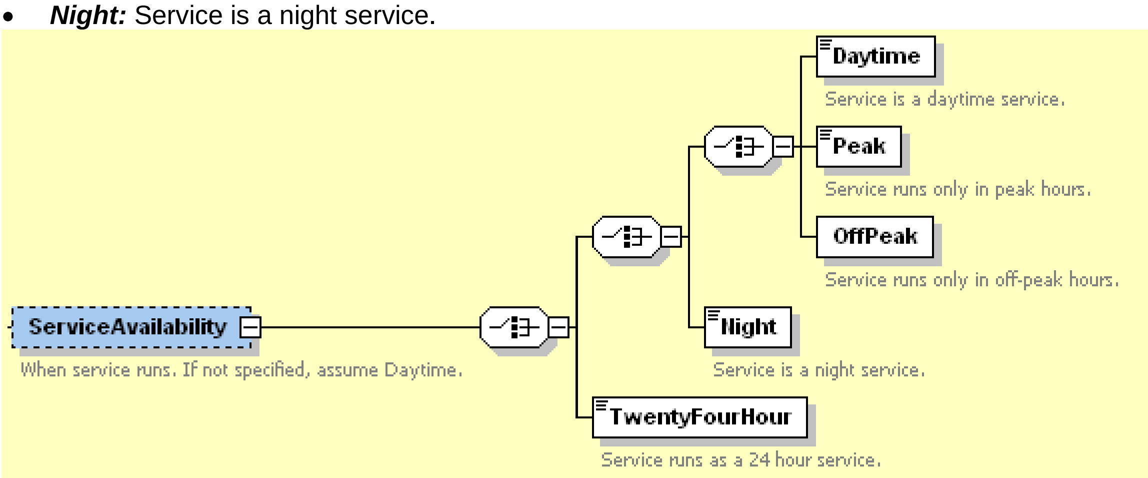


Figure 6-68 – Service / ServiceAvailability Element

6.6.6.12 Service / ToBeMarketedWith Element

The ***ToBeMarketedWith*** element (*Figure 6-69*) records the Services that are normally marketed with the bus service. It contains one or more ***RelatedService*** instances, each of which is an ***AnnotatedServiceRefStructure.***

- ***ServiceRef:*** Reference to another ***Service*** definition provided elsewhere in the document.
- ***Description:*** Text description of the service &/or its identifier if not defined by a service reference.

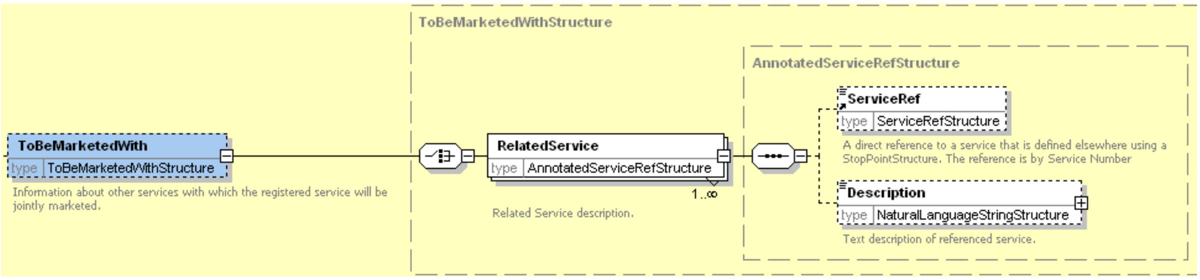


Figure 6-69 – Service / ToBeMarketedWith Element

6.6.7 Service / JourneyGroupings (TXC v2.4)

The ***JourneyGroupings*** element (*Figure 6-70*) allows Journey Groupings for the service to be declared. Each of these may have a description and be associated the vehicle journeys of the service. There are six built in ***JourneyGroupings*** with which journeys will be associated by default. Other Custom Groupings may be added as overrides.
Each ***JourneyGroupings*** has three sets of JourneyGrouping instances:

- **OutboundJourneyGroupings**: Built-in Outbound ***JourneyGroupings***. See below.
- **InboundJourneyGroupings**: Built-in Inbound ***JourneyGroupings***. See below.
- ***Custom*JourneyGroupings**: Built-in Custom ***JourneyGroupings***. See below.

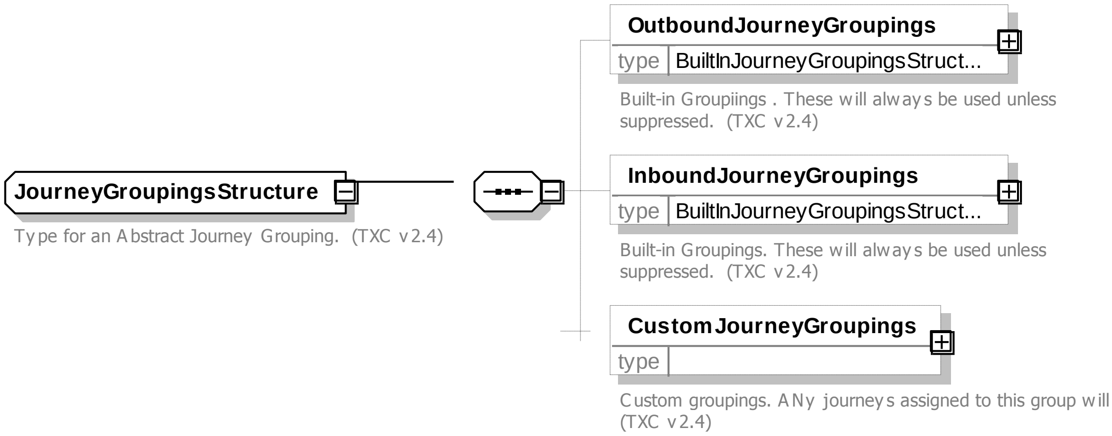


***Figure* 6-70 – Service / JourneyGroupings Element**

6.6.7.1 Service / Built in Journey Grouping Element (TXC v2.4)

The ***BuiltInJourneyGrouping*** Element (Figure 6-71) specifies the properties of an individual built in ***JourneyGrouping***. Each ***JourneyGrouping*** has:

- ***Description:*** Description of ***JourneyGrouping (to be show***n on matrix bed).
- ***Vias:*** : List of Via Names of ***JourneyGrouping*** (to be shown on matrix bed).
  - ***Via***: Name of a place in ***Vias*** list.
- ***StopPointRef***: Designated stop to use to sort columns of ***JourneyGrouping*** by Journey times. (Note yet supported by Publisher – will be ignored if there are Sequence numbers on Journeys corresponding to matrix columns).
- ***Notes***: Notes associated with ***JourneyGrouping***.
- ***Contents:*** : List of Via Names of ***JourneyGrouping*** (to be shown on matrix bed).
  - ***ByOperationalProfile:*** Include any vehicle journeys that have the same day type and direction and are not in a ***CustomJourneyGrouping.***
  - ***None:*** Select journeys as for ***ByOperationalProfile,*** but then suppress this grouping from the published timetable.

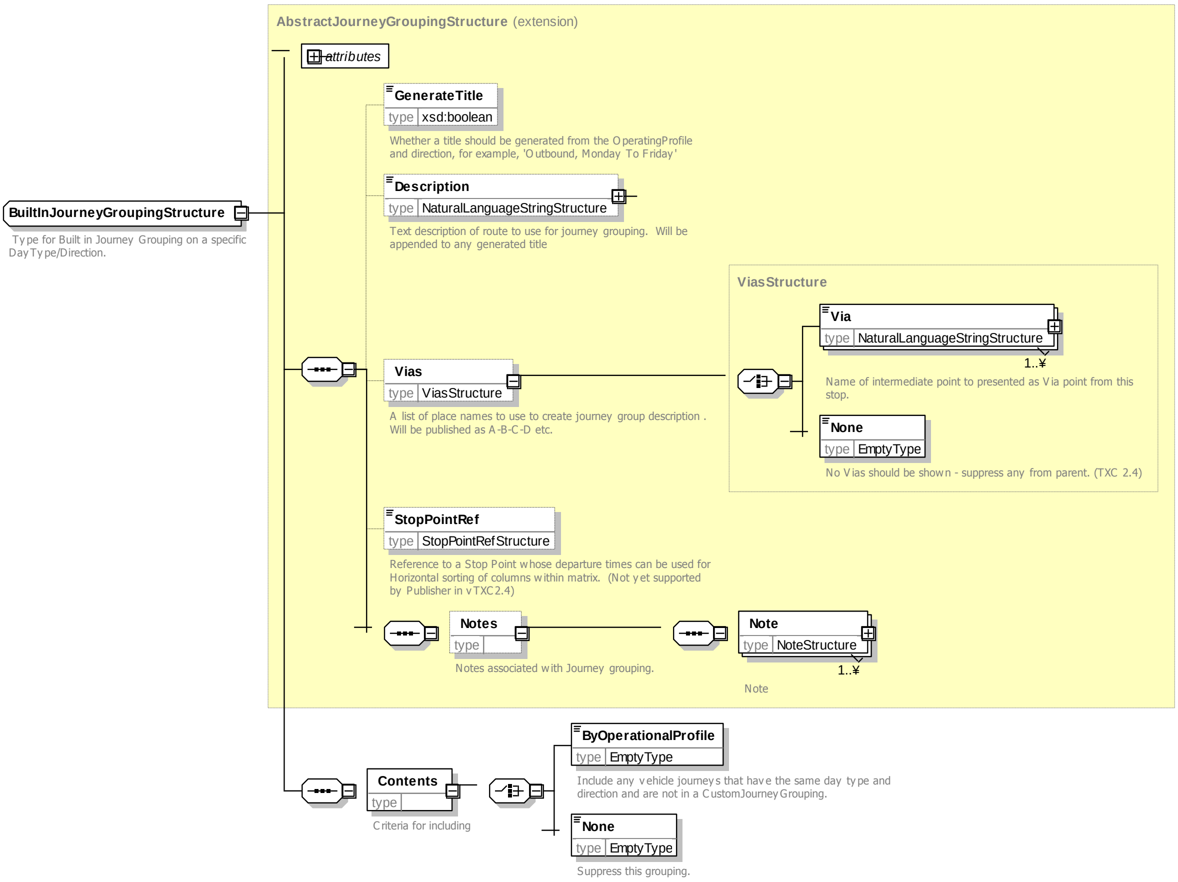


Figure 6-71 – Service / Built in Journey Grouping Element

6.6.7.2 Service / CustomJourneyGrouping Element

The ***CustomJourneyGrouping*** element (*Figure 6-58*) specifies the properties of an individual built in JourneyGrouping. A Custom Journey Group element shares common Properties with a Built-in ***JourneyGrouping***.

In addition, you may specify:

- ***Direction***: Direction of Journeys.
- ***PrivateCode***: Optional preferred colour to use when showing line in graphic media.
- ***VehicleJourneyRefs***: List of references to ***VehicleJourney*** instances that are in the ***CustomJourneyGrouping***.
    - ***VehicleJourneyRef***: Reference to an individual ***VehicleJourney.***

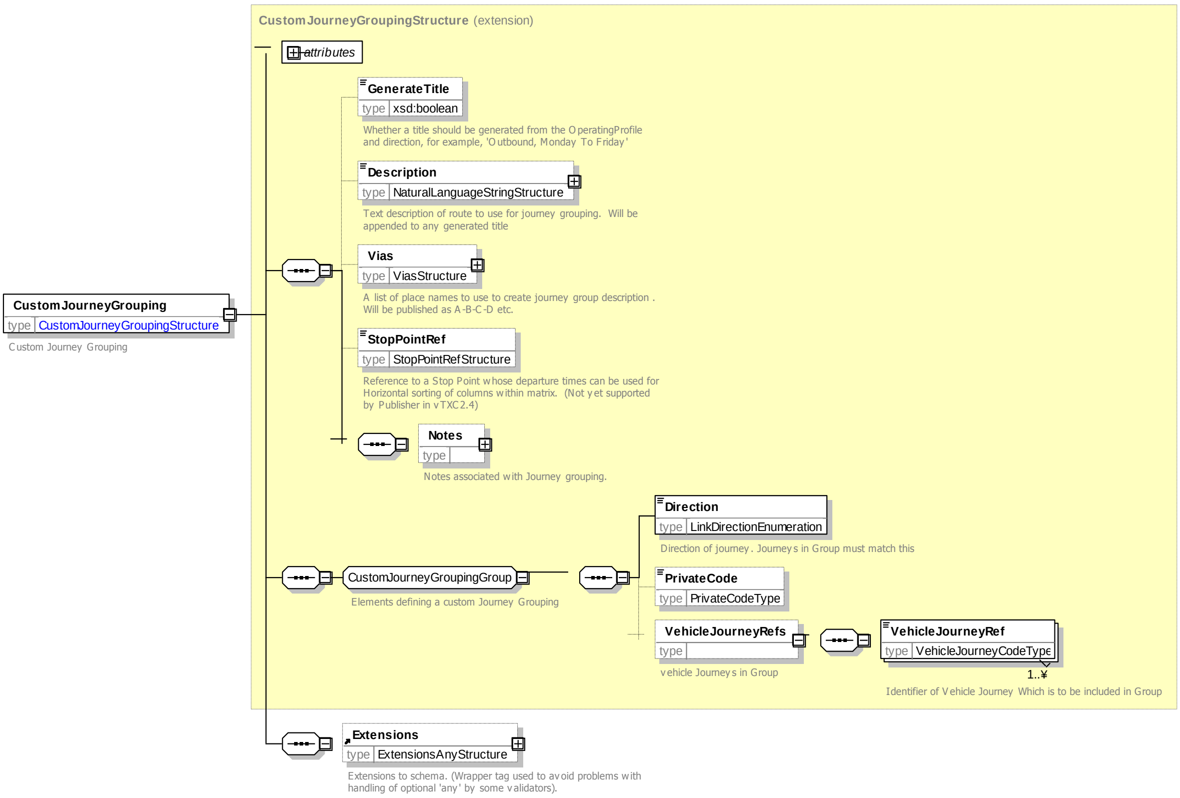


***Figure* 6-72 – Service / CustomJourneyGrouping Element**

6.7 StandardService, JourneyPattern, VehicleJourney

6.7.1 StandardService Element

The ***StandardService*** element (*Figure 6-73*) describes the fixed-route component of a ***Service***. It comprises.

ServicePartGroup – elements common to standard and flexible journey parts.

- ***Origin***: Public name of the place where the service starts.
- ***Destination:*** Public name of the place where the service ends.
- ***Vias:*** Public name(s) of the places that the service route goes past: One or more ***Via*** elements. See below.
- ***UseAllStopPoints:*** Whether the service uses all the stops along its ***Route***.
- ***UseAllStopPoints:*** Whether the service uses all the stops along its ***Route***.
- ***Submode***: Submode for StandardService. See Common Schema Elements.

StandardServiceGroup – elements specific to ***StandardService*** parts.

- ***JourneyPattern***: One or more ***JourneyPattern*** elements representing the working of the service. See below.
- ***Cancellation***: If registration is a cancellation, ***JourneyPattern*** may be omitted.

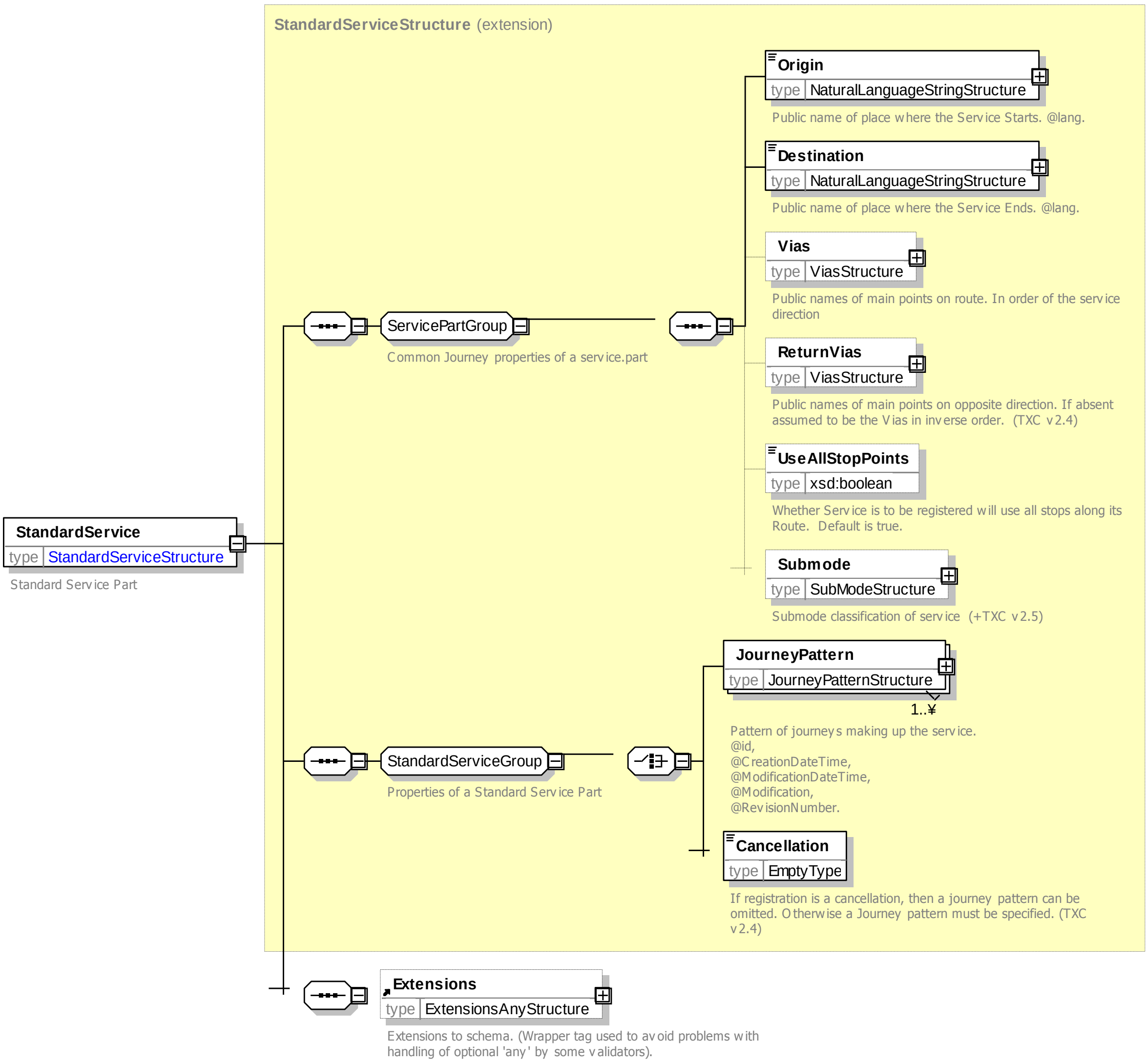


Figure 6-73 – StandardService Element

6.7.2 StandardService / Subelements

6.7.2.1 StandardService / Vias Element

The ***Vias*** element (*Figure 6-58*) allows an indication of the routing of the service as a sequence of text names. It is used to generate the default service description for Matrix beds when publishing the service (reversed for the inbound/outbound directions). It may be overridden (TXC 2.4) using a ***JourneyGrouping.***

Each ***Vias*** has:

- One or more ***Via*** names.
- Or ***None:*** No ***Via*** Names should be shown. (TXC v2.4)

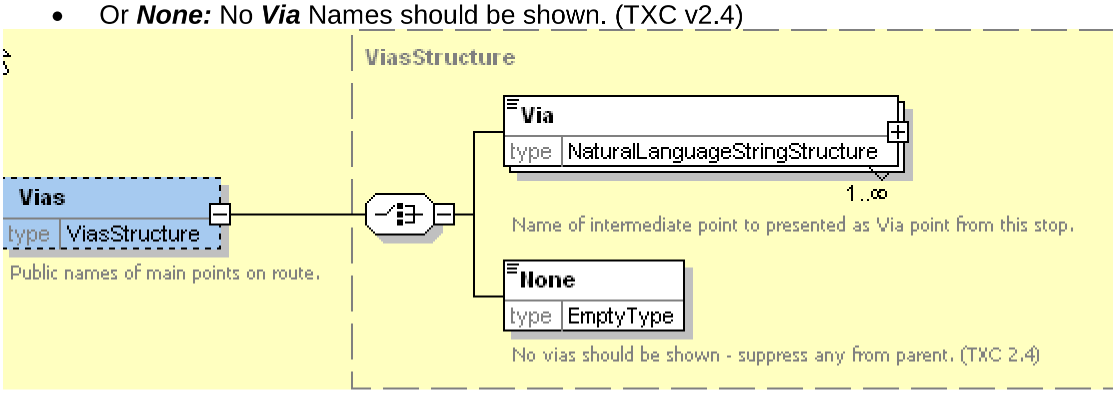


Figure 6-74 – Service / Vias Element

6.7.3 JourneyPatterns

A ***JourneyPattern*** describes a possible bus route of a ***StandardService*** as a sequence of timing links between stops that a vehicle will traverse in a particular order, representing the pattern of working for vehicles of the service.

- Each ***JourneyPattern*** belongs to a ***StandardService***.
- The individual steps of the journey are modelled as ***JourneyPatternTimingLink*** elements; each link has information about the distance to travel, between two stops, and the run time needed. This is to enable the computation of accurate journey timings to each stop. Activity at stop and other information about stop usage is described for each end of the link using ***JourneyPatternStopUsage*** elements.
- The links are grouped into ***JourneyPatternSection*** elements, representing reusable link sequences. Sections are declared within a ***TransXChange*** top-level container element, ***JourneyPatternSections***, and so may be reused in different ***JourneyPattern*** instances.
- The order of ***JourneyPatternTimingLinks*** in each ***JourneyPatternSection,*** and the overall order of the ***JourneyPatternSection*** instances must both follow the order in which they are traversed.
- The timing links of a ***JourneyPattern*** should correspond to the ***RouteLink*** instances of any associated ***Route***, that is be an exact projection on a link-by-link basis of either all the links of route in sequence, or a contiguous subset of the route links in sequence
- In a given ***JourneyPattern,*** the route links of an individual ***RouteSection*** should all be referenced by timing links in a single ***JourneyPatternSection,*** i.e. not be divided between different ***JourneyPatternSection*** instances. A ***JourneyPatternSection*** may however project onto multiple ***RouteSection*** instances.
- A ***JourneyPattern*** may be used in more than one ***VehicleJourney*** on a route. It should be noted that a ***VehicleJourney*** following a ***JourneyPattern*** may not necessarily stop at all stops identified within the ***JourneyPattern***, thus the ***JourneyPattern*** provides the 'super set' of stops of a route, of which all or some may be served by the dependent ***VehicleJourney*** instances. Individual ***VehicleJourney*** instances may subset the full ***JourneyPattern*** stop list either by passing an individual stop, or by short working at either end. They must still follow the route and stop sequence for the part of the journey pattern that they work.

6.7.4 JourneyPattern Element

A ***JourneyPattern*** (*Figure 6-75*) describes the stopping pattern of a standard i.e. fixed route service. A ***JourneyPattern***, is identified by a unique ***id*** attribute, and comprises a number of elements falling into two groups:

1. ***CommonJourneyGroup:*** Shared elements common to journey patterns and vehicle journeys.
2. ***JourneyPatternGroup:*** Elements specific to journey patterns.

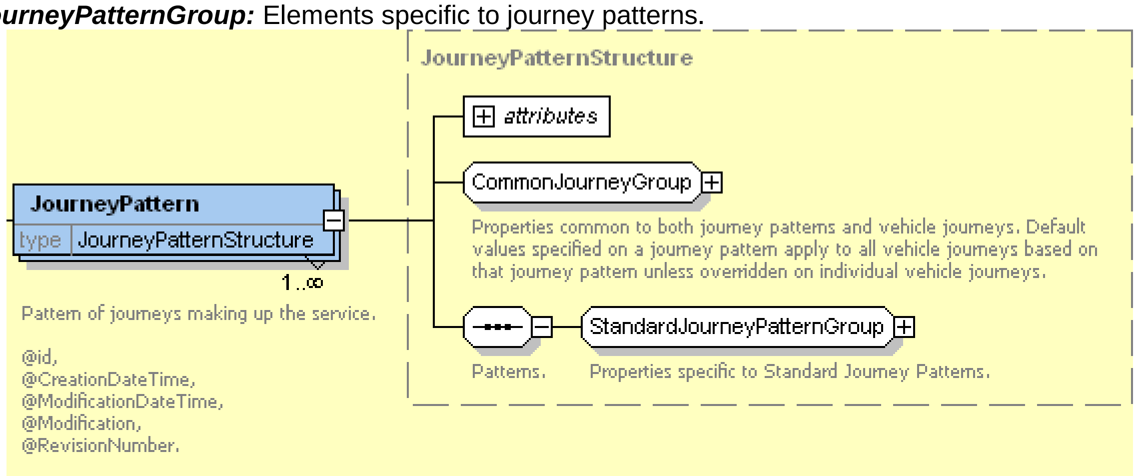


Figure 6-75 – JourneyPattern Element

6.7.4.1 JourneyPattern / CommonJourneyGroup

The ***CommonJourneyGroup*** (*Figure 6-76*) holds identity and operational information that is common to both a ***JourneyPattern*** and a ***VehicleJourney:*** the ***JourneyPattern*** instances provide default values to use on dependent ***VehicleJourney*** instances if no specific override is provided on the ***VehicleJourney***.

- ***JourneyIdentificationGroup***: Elements for identifying the journey – see below.
- ***OperatingProfile:*** Specifies operational days and times associated with the JourneyPattern. If not specified inherited from ***Service***.
- ***Operational:*** Specifies additional operational information associated with the journey. See below. Normally this is not required since it is the same as for the service. Includes ***TicketMachine*** and ***Block*** elements. See below.
- ***TimeDemand***: Classification of the route as to when peak demand occurs. See *Table 6-39*.

Value	Description
earlyMorning	Early Morning.
offPeak	Off Peak.
peakMorning	Peak Morning.
peakAfternoon	Peak Afternoon.
evening	Evening.
lateEvening	Late Evening.
saturdayMorning	Saturday Morning.
saturdayDaytime	Saturday Daytime.
saturdayEvening	Saturday Evening.
sunday	Sunday.
bankHoliday	Bank Holiday.

Table 6-39 – Allowed Values for TimeDemand

- ***CommercialBasis:*** on which the service is offered. May be overridden for specific journey patterns, vehicle Journeys and timing links. See Table 6-40 for allowed values.

Value	Description
contracted	Service on Link is contracted
notContracted	Service on Link is not contracted.
partContacted	Service on Link is part contracted.
inherit	Basis is same as that of parent service or journey pattern
unknown	Basis is unknown.

Table 6-40 – Allowed Values for Service / CommercialBasis.

- ***LayoverPoint***: Points at which the service lays over. See below.

- ***GarageRef***: A garage from which the ***Service*** operates.

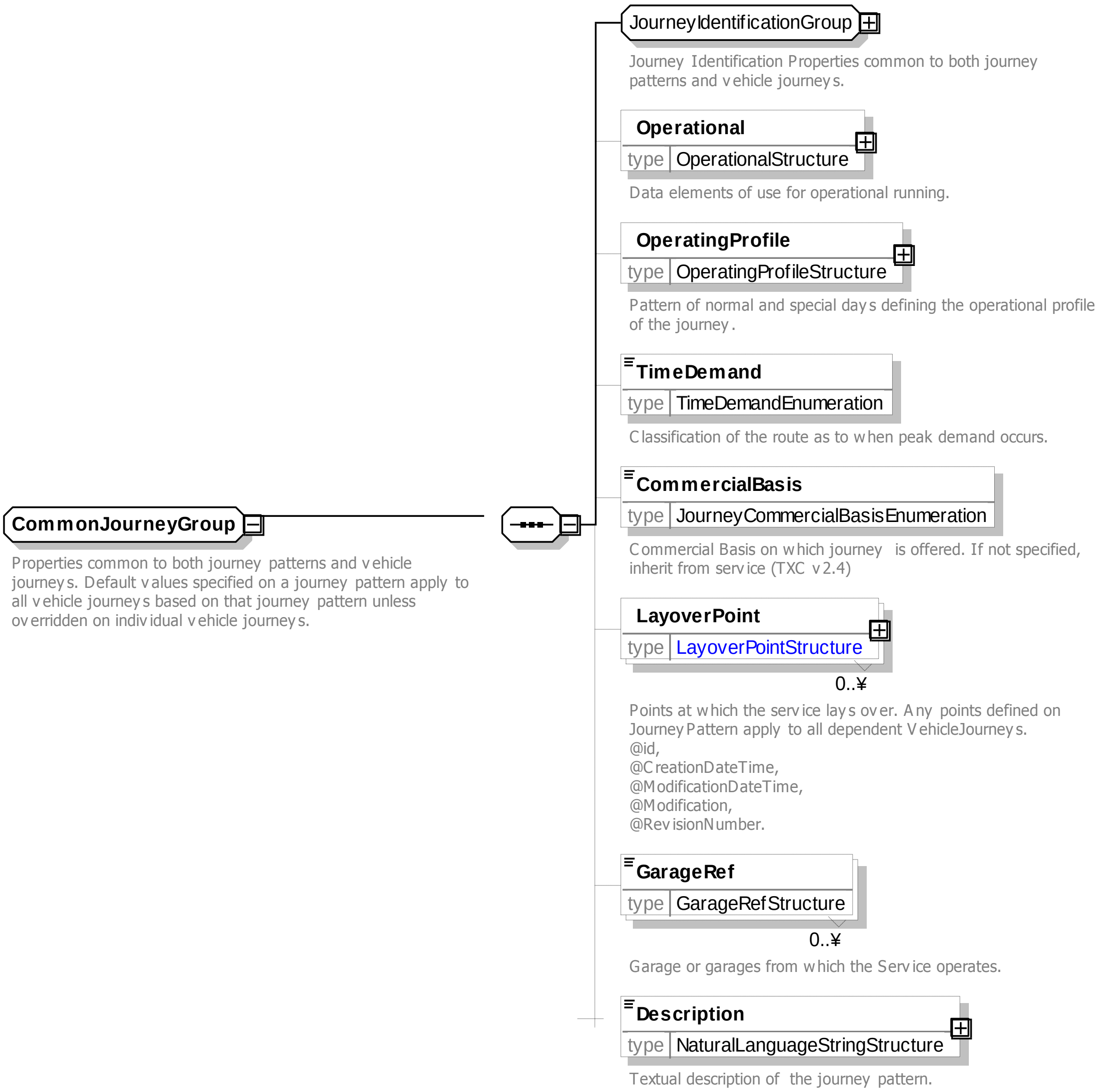


Figure 6-76 – JourneyPattern / CommonJourneyGroup

6.7.4.2 JourneyPattern / JourneyIdentificationGroup

The JourneyIdentificationGroup holds information for identifying a journey.

- ***RouteRef***: The ***Route*** which the ***JourneyPattern*** follows. See ***Route*** above.
- ***JourneyPatternSectionRefs:*** An ordered collection of references to ***JourneyPatternSections (***as ***JourneyPatternSectionRef*** instances), that contain the journey pattern timing links making up the ***JourneyPattern***. See ***JourneyPatternSection*** later.

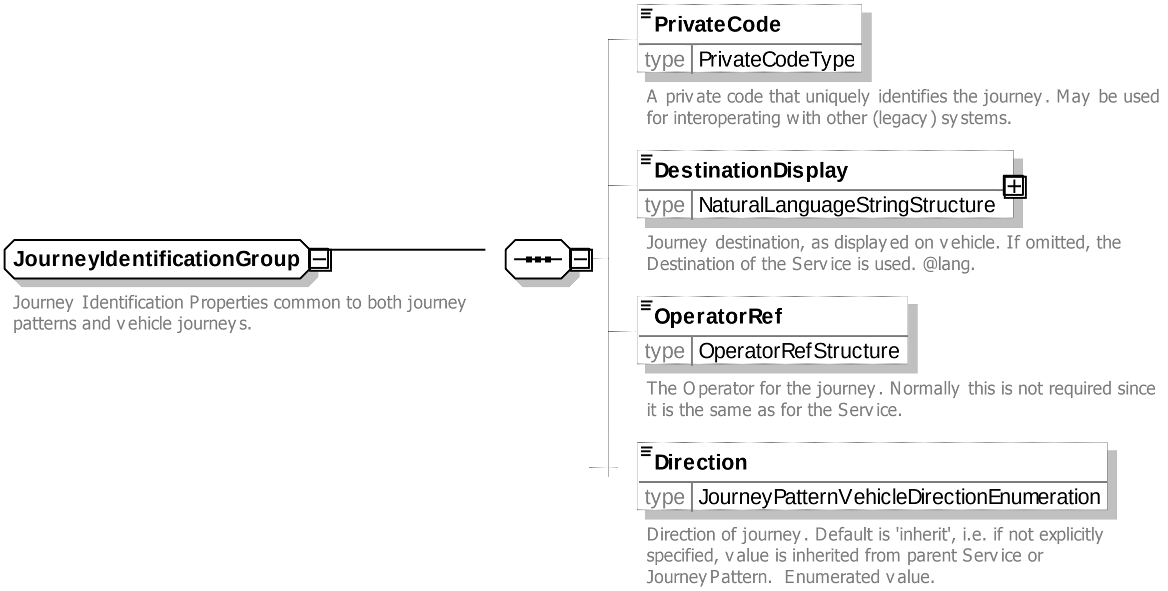


Figure 6-77 – JourneyIdentificationGroup

6.7.5 JourneyPattern Subelements

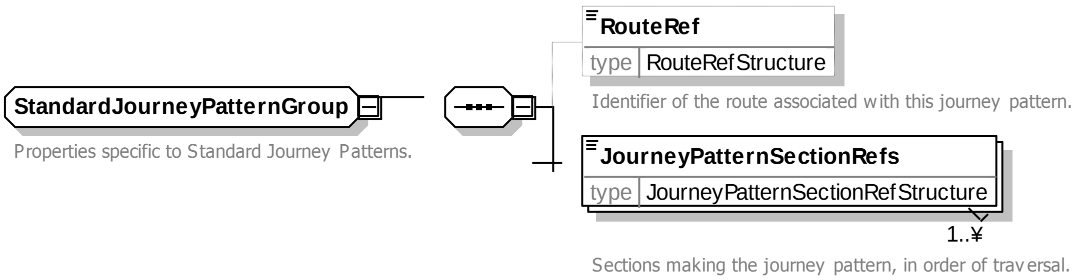
6.7.5.1 JourneyPattern / StandardJourneyPatternGroup

The ***StandardJourneyPatternGroup*** (Figure 6-77) holds information to identify a journey***:***

- ***PrivateCode***: A unique private code that can be used to identify the ***JourneyPattern***.
- ***DestinationDisplay***: Journey destination, as displayed on vehicle. If omitted, the ***Destination*** of the ***Service*** is used.
- ***OperatorRef:*** The operator for the journey. Normally this is not required since it is the same as for the service.
- ***Direction***: The default ***Direction*** of the ***JourneyPattern.*** Default is '*inherit'*. See Table 6-41 ***f***or allowed values***.***

Value	Description
inherit	Use value from Service.
inbound	Inbound Direction.
outbound	Outbound Direction.
clockwise	Clockwise Direction.
antiClockwise	Anti-Clockwise Direction.

Table 6-41 – Allowed Values for JourneyPattern / Direction



.

Figure 6-78 – JourneyPattern / StandardJourneyPatternGroup

6.7.5.2 CommonJourneyGroup / Operational / OperationalVariant Element

The ***OperationalVariant*** element (*Error! Reference source not found.*) specifies operational information variants for a particular day associated with the ***JourneyPattern***:

- Day on which the variant applies specified either as:
  - ***OperationalProfile***: In line description of day properties

- ○ ***DayTypeRef***: Reference to a ***DayType***. See Calendar/DayType later below.
- ***OperationalGroup***: Data elements that relate to operational aspects of the journey: as above.

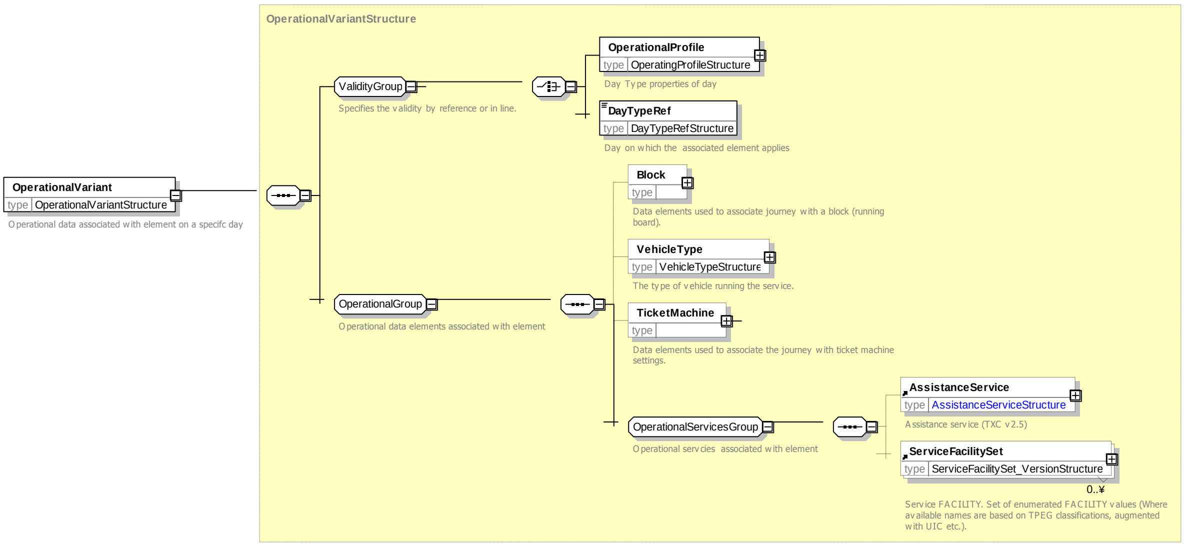


Figure 6-79 – JourneyPattern / Operational / OperationalVariant Element

6.7.5.3 CommonJourneyGroup / Block Element

The ***Block*** element (*Figure 6-80*) specifies information about the block (running board) of a journey. A block enables ***VehicleJourney*** instances to be assigned to a logical group of journeys that will be carried out by the same vehicle.

- ***Description***: Text describing the block.
- ***BlockNumber:*** The number of the block associated with the journey. ***VehicleJourney*** instances with the same ***BlockNumber*** will be carried out by the same vehicle
- ***Note***: Explanatory text to explaining any further operational particulars about the block.

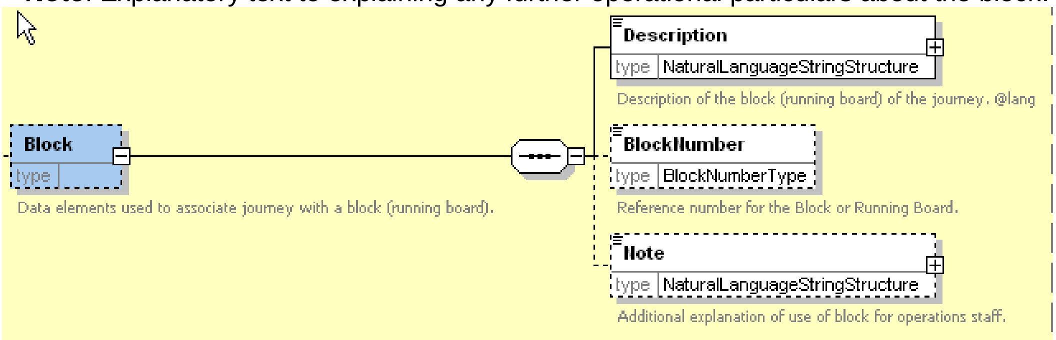


Figure 6-80 – Operational / Block Element

6.7.5.4 CommonJourneyGroup / VehicleType Element

The ***VehicleType*** element (*Figure 6-81*) describes a type of vehicle running a service.

- ***VehicleTypeCode***: Arbitrary code that classifies the vehicle.
- ***Description***: Free text description of vehicle type.
- ***VehicleEquipment***: Different types of equipment available on the Vehicle.
  - ○ ***PassengerInfoEquipment***: Information relating to passenger Information facilities on the vehicle. (+TXC V2.4)
  - ○ ***AccessEquipment***: Information relating to data access on the vehicle. (+TXC V2.4)
  - ○ ***WheelchairEquipment***: Information relating to wheelchair access on the vehicle. (+TXC V2.4)

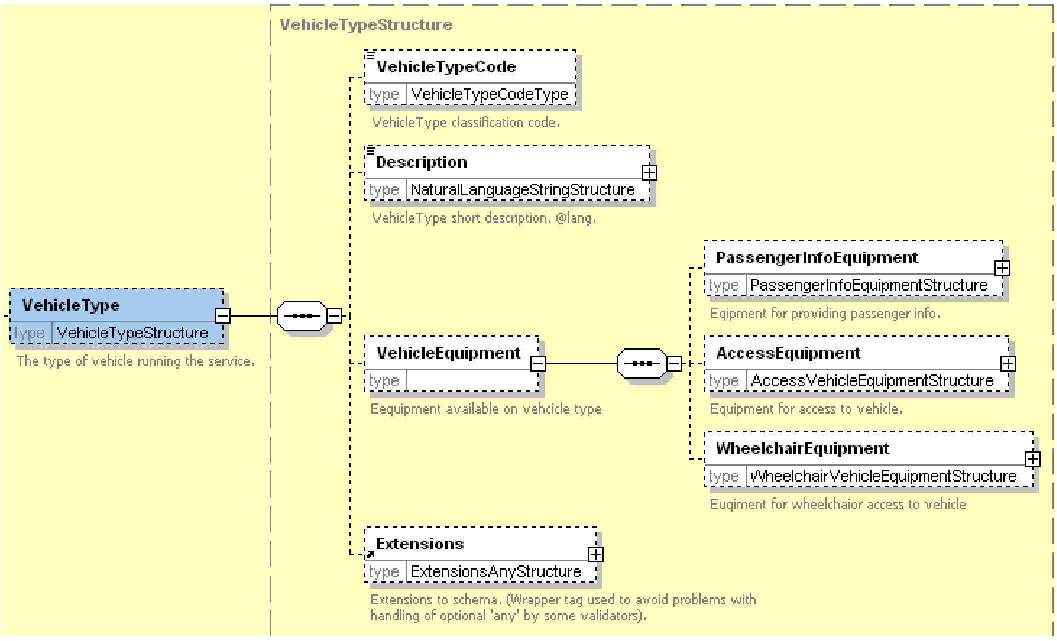


Figure 6-81 – Operational / VehicleType Element

6.7.5.5VehicleType / Equipment Element (+TXC V2.5)

Detailed properties of on vehicle services are described with different types of Equipment. All types of equipment share common ***Equipment*** element properties(*Figure 6-83*) (+TXC V2.5)

- ***Name***: Name of Equipment (+TXC v2.5).
- ***Availability***: Condition specifying when equipment is available (+TXC v2.5).
  - ***DaysOfWeek***. Which days of week equipment is available
  - ***Timebands:*** When equipment is available.

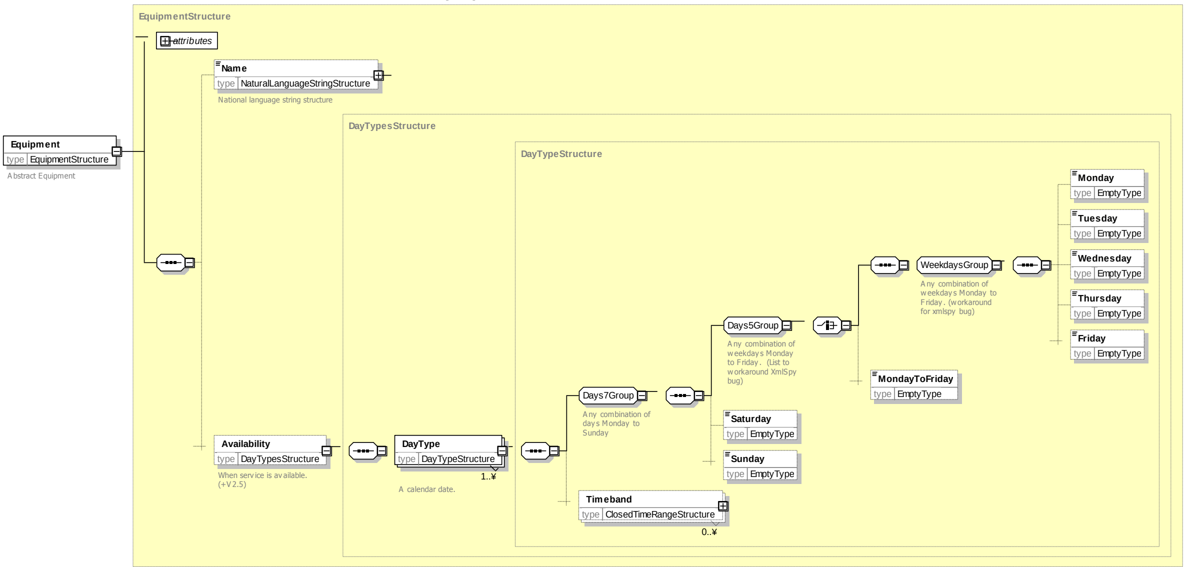


Figure 6-82 – Operational / Abstract Equipment Element

6.7.5.6VehicleType / PassengerInfoEquipment Element (+TXC V2.4)

The ***PassengerInfoEquipment*** element (*Figure 6-83*) describes a type of vehicle equipment relating to passenger information service. (+TXC V2.4)

- ***PassengerInfo***: Arbitrary code that classifies the vehicle. See Table 6-40 for allowed values.

Value	Description
nextStopIndicator	Vehicle has Next Stop Indicator

stopAnnouncements	Vehicle has Stop Announcements
passengerInfoFacility	Vehicle has Passenger Info Facility on-board
other	other

Table 6-42 – Allowed Values for PassengerInfoEquipment / PassengerInfo.

- ***AccessibilityInfo***: Arbitrary code that classifies the vehicle. See Table 6-40 for allowed values.

Value	Description
audioInformation	Vehicle has audioInformation
audioForHearingImpaired	Vehicle has audioForHearingImpaired
visualDisplays	Vehicle has visualDisplays
displaysForVisuallyImpaired	Vehicle has displaysForVisuallyImpaired
tactilePlatformEdges	Vehicle has tactilePlatformEdges
tactileGuidingStrips	Vehicle has tactileGuidingStrips
largePrintTimetables	Vehicle has largePrintTimetables
other	other

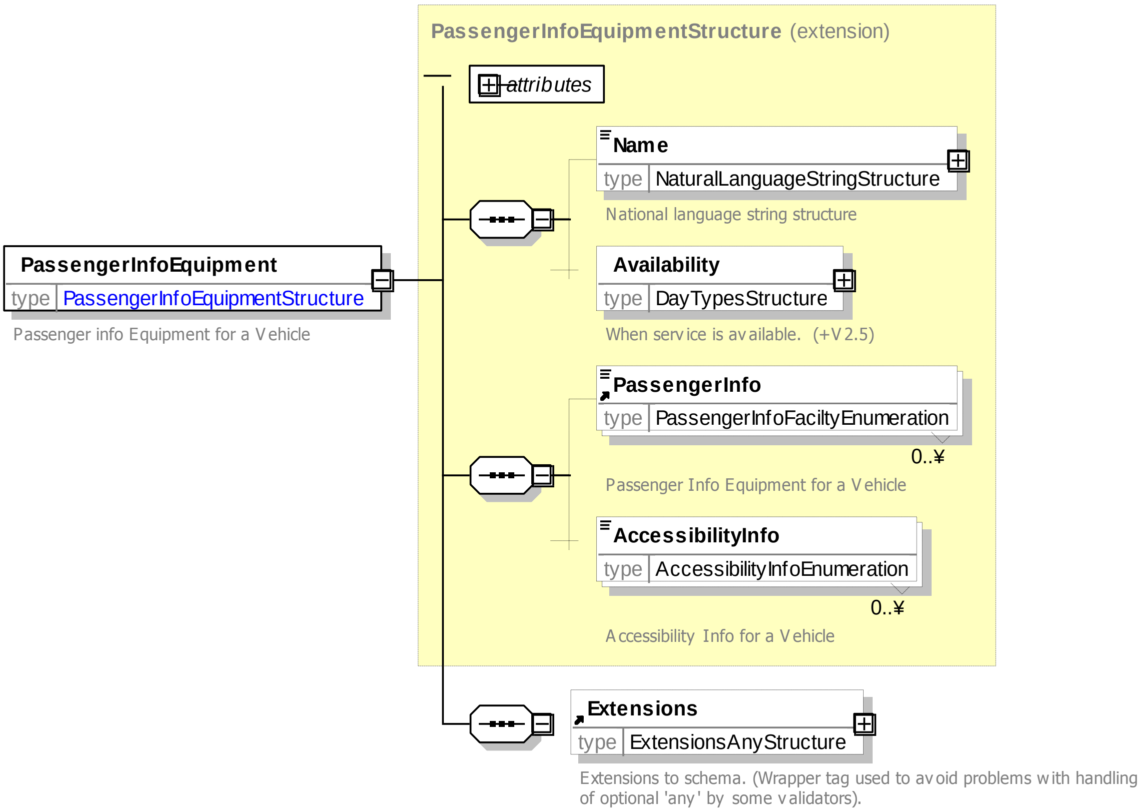


Figure 6-83 – Operational / Equipment Element

6.7.5.7 VehicleType / AccessVehicleEquipment Element (+TXC V2.4)

The ***AccessVehicleEquipment*** element (Figure 6-84Figure 6-84) describes a type of vehicle equipment relating to physical access to the vehicle relevant to accessibility planning. (+TXC V2.4)

- ***LowFloor***: Whether Vehicle has low floor access.
- ***Hoist***: Whether Vehicle has hoist for wheelchair access. (+TXC v2.5)
- ***Ramp***: Whether Vehicle has deployable ramp.
- ***RampBearingCapacity***: Whether Vehicle has deployable ramp.
- ***NumberOfSteps***: Number of steps to board vehicle.
- ***BoardingHeight***: Rise from ground needed for access to vehicle floor.
- ***GapToPlatform***: Gap between vehicle and platform (May vary by stop).
- ***WidthOfAccessArea***: Access width for entry.
- ***HeightOfAccessArea***: Access height for entry.
- ***AutomaticDoors***: Whether Vehicle has automatic doors.

- ***SuitableFor:*** Mobility need for which access is suitable. See Table 6-43 below. (+TXC v2.5)

Value	Description
wheelchair	Wheelchair
assistedWheelchair	Wheelchair pushed by companion
motorisedWheelchair	Motorised Wheelchair
mobilityScooter	Small mobility Scooter: A Class 2 scooter under the CPT classification with 3 or 4 wheels, not more than 600mm wide and 1000 mm long and with a turning radius not exceeding 1200mm. Normally weigh about 65 kg.
roadMobilityScooter	Large Mobility Scooter: A Class 3 scooter under the CPT classification. Class 3 scooters are bigger and have light for road use. They are not normally allowed on buses.
walkingFrame	Walking Frame
restrictedMobility	Restricted Mobility
normal	Normal mobility

Table 6-43 – Allowed Values for *MobilityNeed*

- ***AssistanceNeeded:*** Nature of assistance needed to board See (Table 6-44). (+TXC v2.5)

Value	Description
levelAccess	Level access – passenger can propel themselves
rampRequired	Assistance with ramp needed.
hoistRequired	Assistance with hoist needed.
unknown	Not known.

Table 6-44 – Allowed Values for *AssistanceNeeded*

- ***AssistedBoardingLocation:*** Whether boarding has to be done at a specific position on the platform. See Table 6-45). (+TXC v2.5)

Value	Description
boardAtAnyDoor	Boarding can be at any location
boardOnlyAtSpecifiedPositions	Boarding must be at specific positions on platform
unknown	Not known.

Table 6-45 – Allowed Values for *AssistedBoardingLocation*

- ***GuideDogsAllowed:*** Whether guide dogs are allowed. (+TXC v2.5)

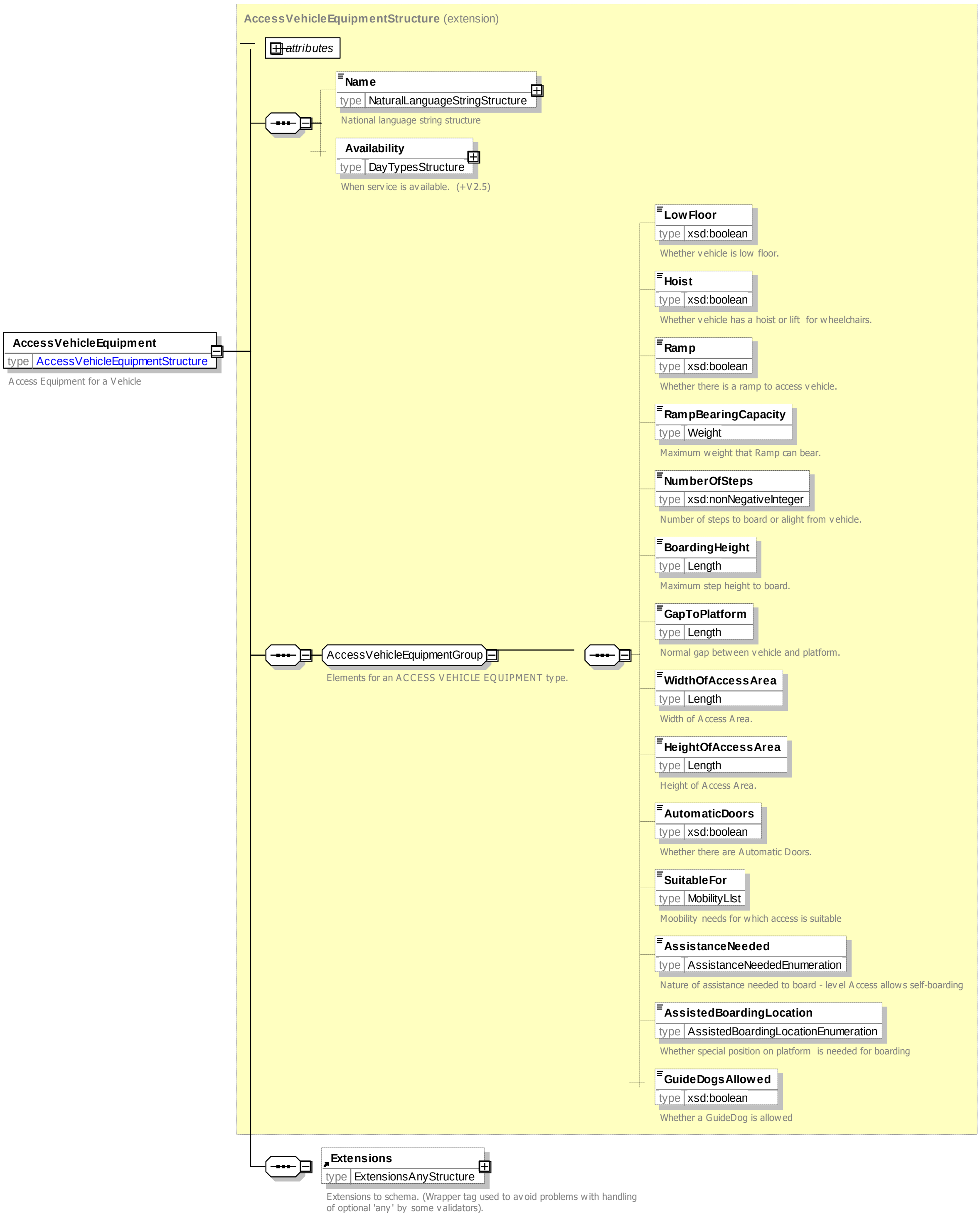


Figure 6-84 – VehicleType/ AccessVehicleEquipment Element (+TXC v2.4)

6.7.5.8 VehicleType / WheelchairEquipment Element (+TXC V2.4)

The ***WheelchairEquipment*** element *(*Figure 6-85) describes a type of on board vehicle equipment for wheelchair access on the vehicle. (+TXC V2.4)

- ***NumberOfWheelChairAreas***: Number of available wheelchair areas in vehicle.
- ***WidthOfAccessArea***: Access width available for entry by wheelchair.
- ***HeightOfAccessArea***: Access height available for entry by wheelchair.
- ***WheelchairTurningCircle***: Wheel chair turning circle on board.

- ***SuitableFor:*** Mobility need for which access is suitable, for example motorisedWheelchair, mobilityScooter, etc., See Table 6-43 earlier. (+TXC v2.5.)
- ***BookingRequired:*** Whether booking is required for wheelchair use. (+TXC v2.5.)
- ***BookingNumber:*** Phone number to call for wheelchair booking (+TXC v2.5.)

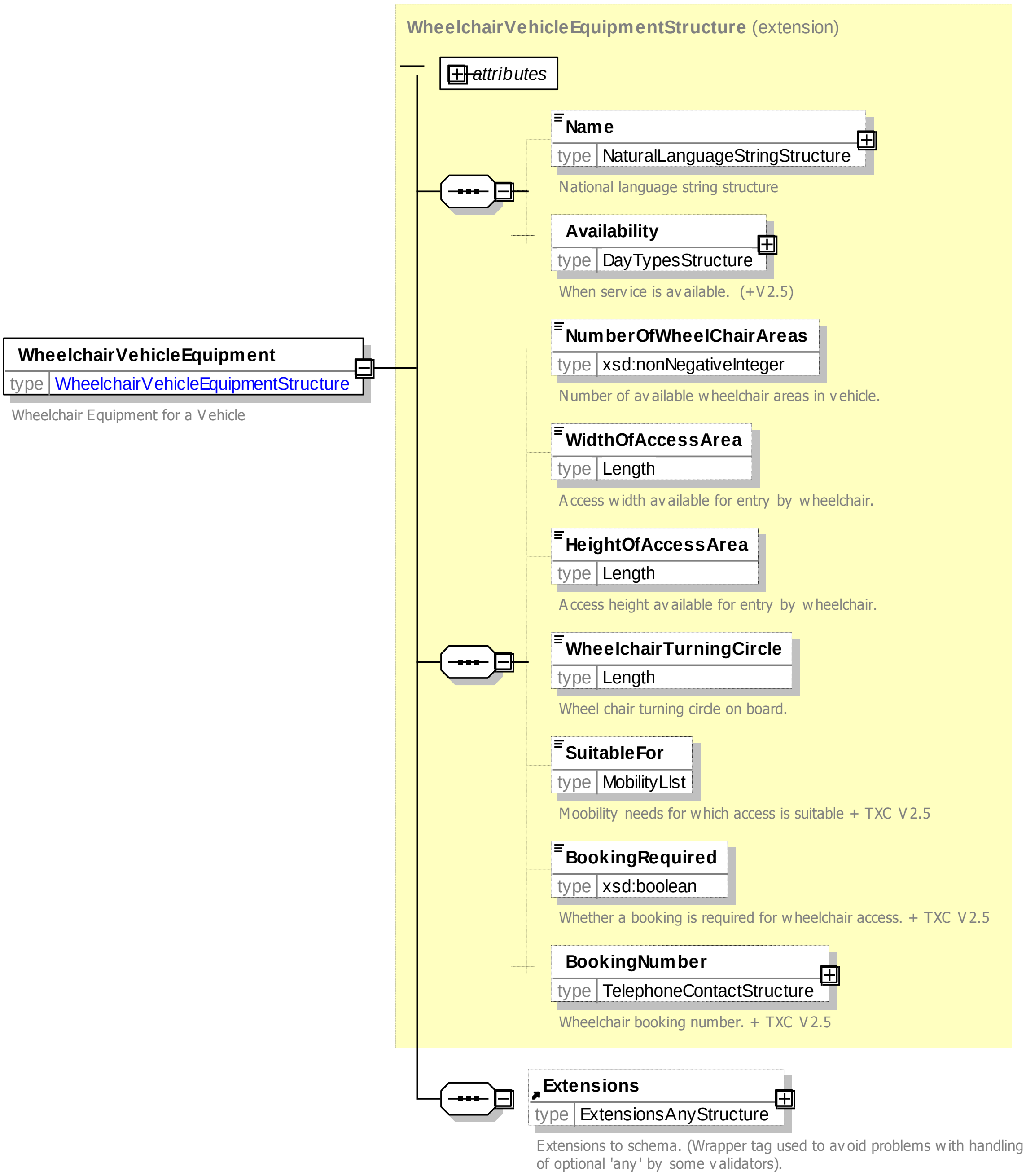


Figure 6-85 – VehicleType/ WheelchairVehicleEquipment Element (+TXC v2.4)

6.7.5.9CommonJourneyGroup / Operational / TicketMachine Element

The ***TicketMachine*** element (Figure 6-86) specifies information for associating a journey with the settings of a ticket machine***.***

- ***TicketMachineServiceCode:*** Unique Identifier associated with service for use in ticketing machine systems. If not specified, defaults to any value specified at the Service Level.
- ***JourneyCode***: The identifier used by the ticket machine system to refer to the journey.
- ***Direction***: The direction used by the ticket machine system to refer to the journey.

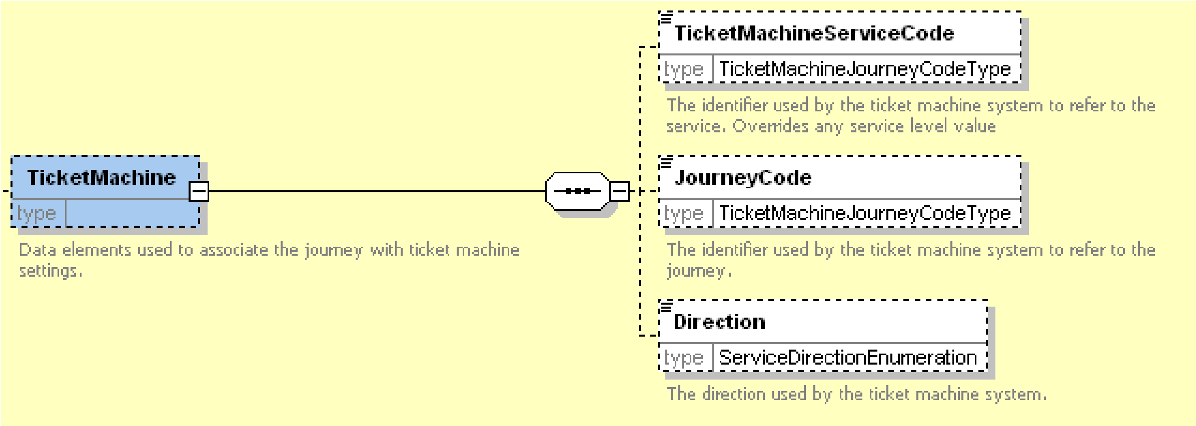


Figure 6-86 – CommonJourneyGroup /Operational / TicketMachine Element

6.7.5.10 CommonJourneyGroup / LayoverPoint Element

The ***LayoverPoint*** element (*Figure 6-87*) describes a layover point used in a journey pattern. It is identified by an ***id*** attribute, and comprises:

- ***Duration***: Time of wait at layover point. Uses standard duration type.
- ***Name***: Free text description of layover point.
- ***Location***: Location of layover point.
- ***Minimum Duration:*** Minimum time for a layover at a point (+v2.4)

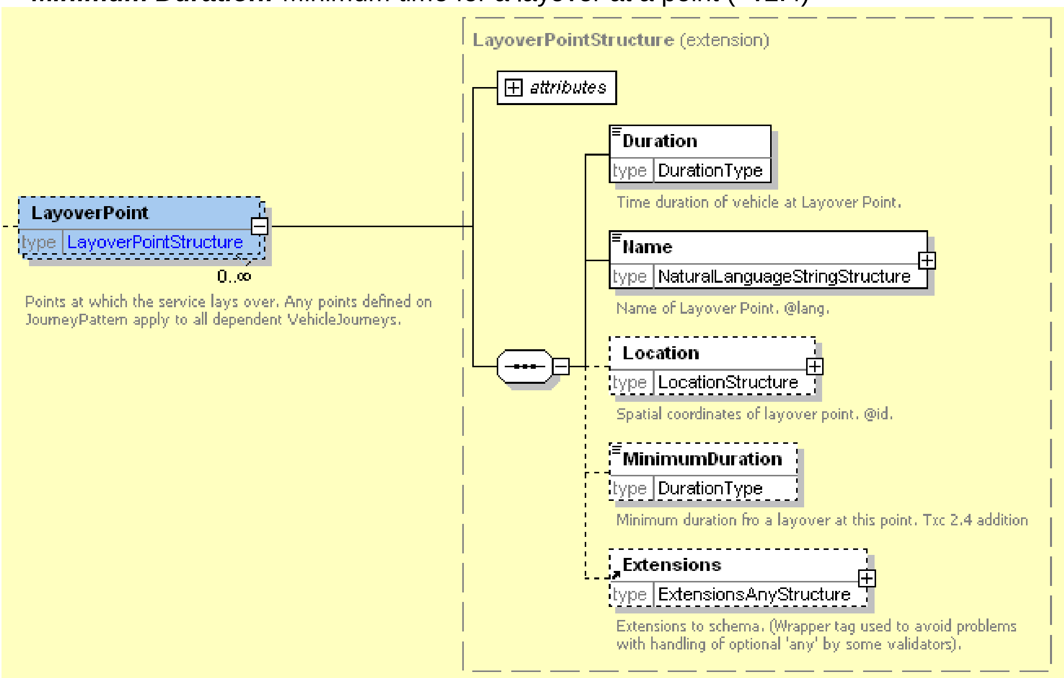


Figure 6-87 – JourneyPattern / LayoverPoint Element

6.7.6 JourneyPatternSection Element

A ***JourneyPatternSection*** (*Figure 6-88*) declares and groups an ordered collection of ***JourneyPatternTimingLink*** elements. Each ***JourneyPatternSection*** can be identified by a unique ***id*** attribute.

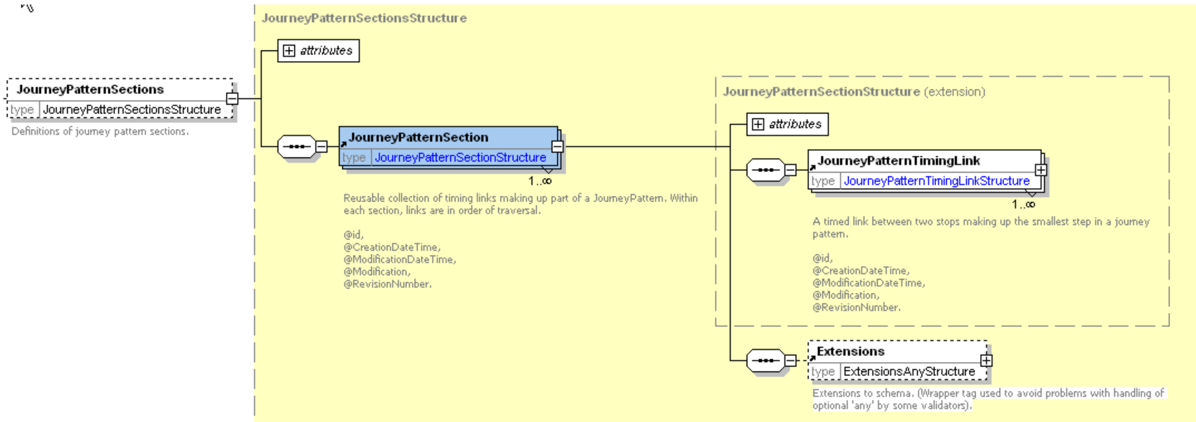


Figure 6-88 – JourneyPatternSection Element

6.7.7 JourneyPatternTimingLink Element

A ***JourneyPatternTimingLink*** (*Figure 6-89*) describes a timed link connecting two stops of a ***JourneyPattern*** of a ***StandardService.*** It allows a precise default timing to be set for traversing a link. Each ***JourneyPatternTimingLink*** can be identified by a unique ***id*** attribute, and comprises a number of elements falling into two groups:

1. ***CommonTimingLinkGroup:*** Shared elements common to journey pattern timing links and to vehicle journey timing links.
2. ***JourneyPatternTimingLinkGroup:*** Elements specific to journey pattern timing links.

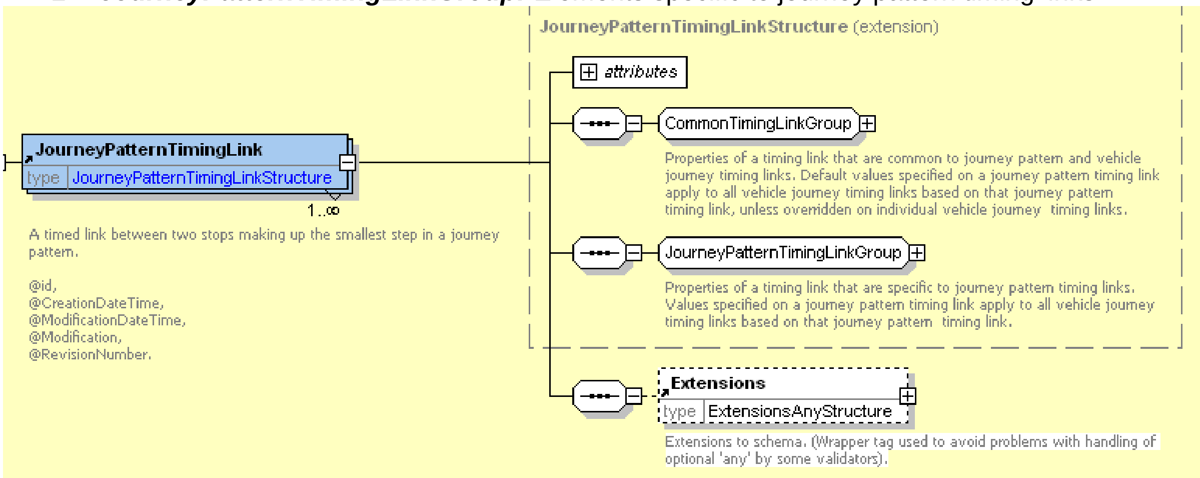


Figure 6-89 – JourneyPatternTimingLink Element

6.7.7.1 JourneyPatternTimingLink / CommonTimingLinkGroup

The ***CommonTimingLinkGroup*** (*Figure 6-91*) holds elements that are common to both a ***JourneyPatternTimingLink*** and a ***VehicleJourneyTimingLink;*** the ***JourneyPatternTimingLink*** instances provide default values to use on dependent ***VehicleJourneyTimingLink*** instances if no specific override is provided on a particular ***VehicleJourneyTimingLink***.

- ***JourneyPattern Ref:*** Optional reference to parent ***JourneyPattern***. Normally not stated as given by containing context, but may be specified when using timing link as a stand-alone artefact. If given by context, this value is ignored. (TXC v2.4)
- ***HailAndRide:*** Whether link operates as a Hail and Ride service. Normally stops at both ends of a link flagged as ***HailAndRide*** will be ***HailAndRide*** stops.
- ***Express:*** Whether link operates as an express section (that is, typically going past a stop without stopping at one or both ends of the link).
- ***StoppingArrangements*** Text description of facilities/requirements for stopping associated with link.

- ***DutyCrewCode:*** Code identifying duty crew operating bus over link. Note that if used, a value need not be specified on every link of a journey pattern: any value specified is assumed to run for all intervening links until the next link with a value is encountered. (NB this element should more correctly be called ***DutyCrewRef***).
- ***CommercialBasis:*** on which this link of the service is offered. See for allowed values.

Value	Description
contracted	Service on Link is contracted
notContracted	Service on Link is not contracted.
inherit	Basis is same as parent vehicle Journey or Journey Pattern
unknown	Basis is unknown.

Table 6-46 – Allowed Values for TimingLink / CommercialBasis

-

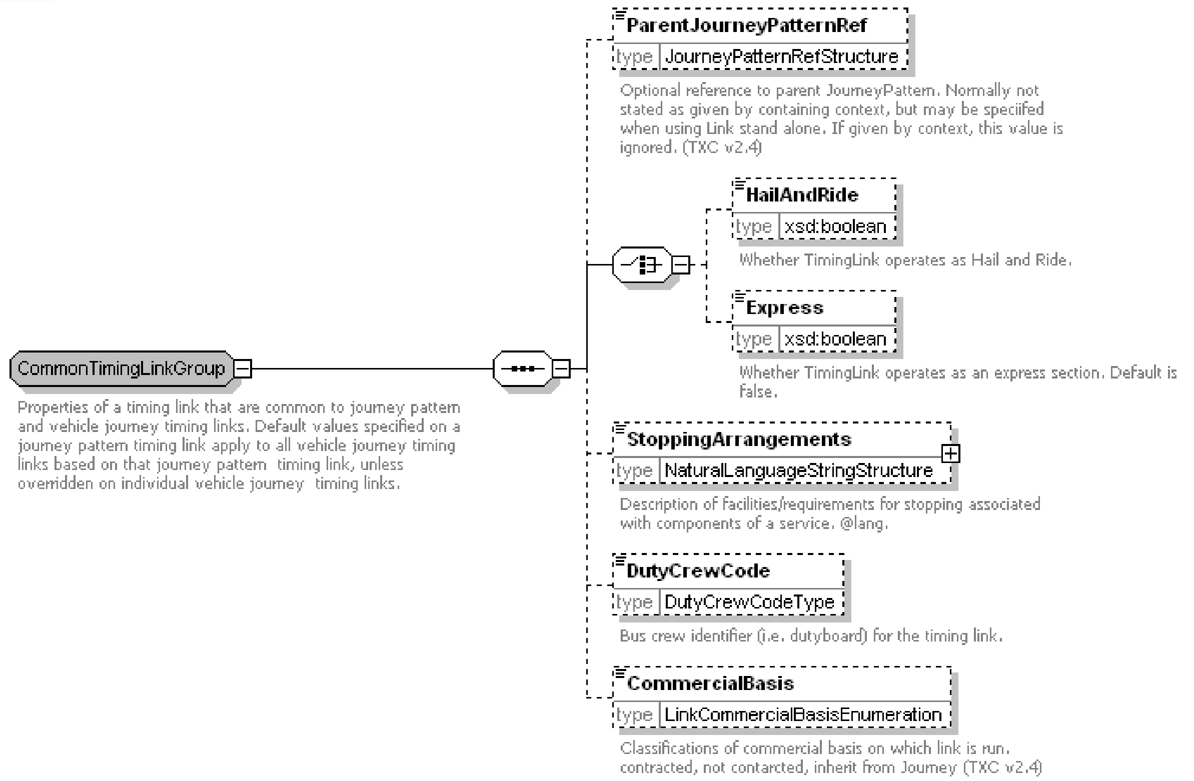


Figure 6-90 – JourneyPatternTimingLink / CommonTimingLinkGroup

6.7.7.2 JourneyPatternTimingLink / JourneyPatternTimingLinkGroup

The ***JourneyPatternTimingLinkGroup*** (*Figure 6-91*) holds elements that are specific to a ***JourneyPatternTimingLink:***

- ***From:*** Default usage details of from stop, specified by a ***JourneyPatternStopUsageStructure.*** See later.
- ***To:*** Default usage details of from stop, specified by a ***JourneyPatternStopUsageStructure*** element. See later.
- ***RouteLinkRef:*** Optional reference to a ***RouteLink*** onto which timing link projects.
- ***Direction:*** Direction of link. Default is '*inherit'*. See *Table 6-47.*
- ***RunTime:*** Time taken to traverse link. Normally this will be greater than zero. A ***RunTime*** value should only be zero, i.e. *PT0M*, where stops are very close together and the time between them would be less than a minute (even then it is desirable to give the value in seconds if known, e.g. *PT0M35S*). A journey pattern should never have zero minute timings for all its links.

Value	Description
inherit	Use value from Journey Pattern.
inbound	Inbound Direction.

outbound	Outbound Direction.
clockwise	Clockwise Direction.
antiClockwise	Anti-Clockwise Direction.

Table 6-47 – Allowed Values for VehicleJourney / Direction

- ***Distance:*** Distance along link path in metres.

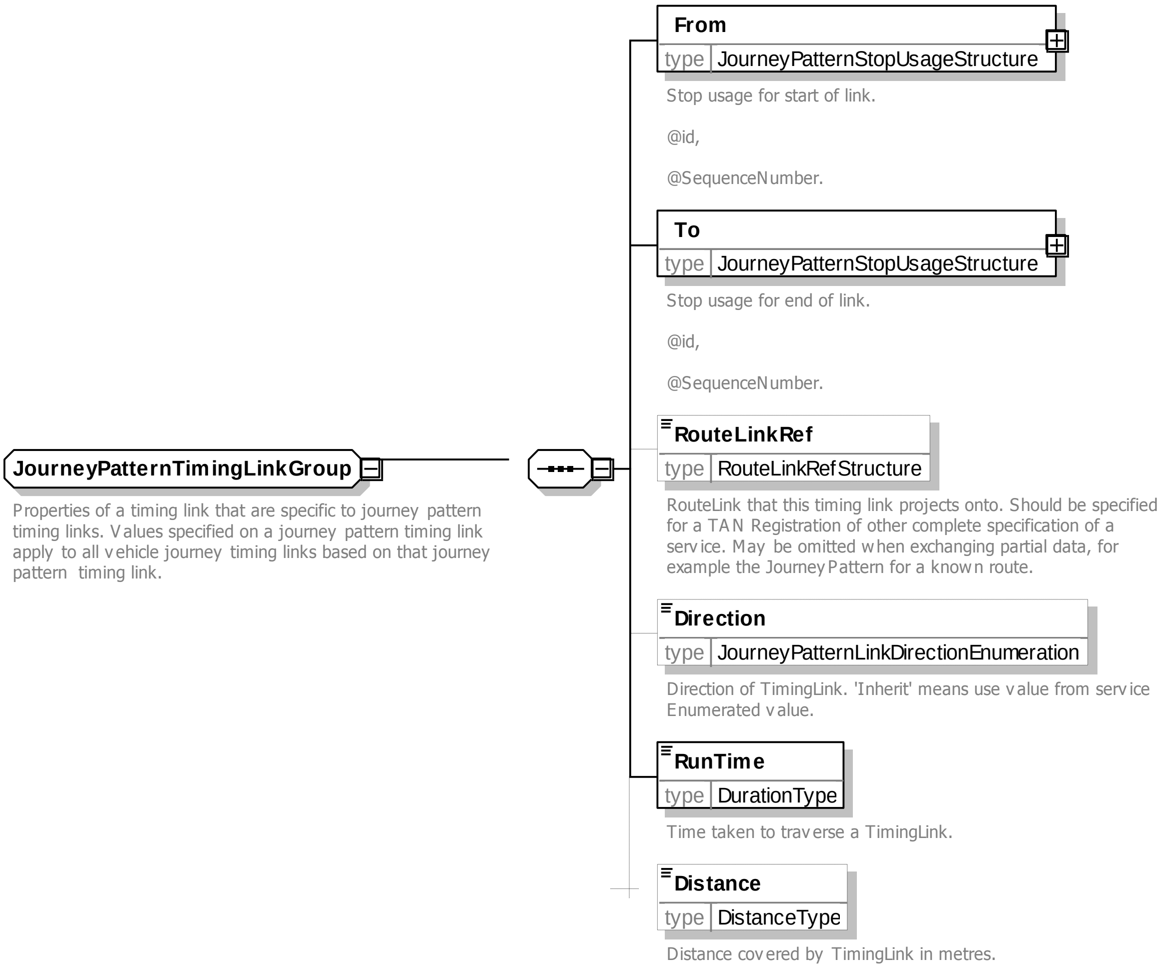


Figure 6-91 – JourneyPatternTimingLink / JourneyPatternTimingLinkGroup

6.7.8 JourneyPatternStopUsageStructure

The ***JourneyPatternStopUsageStructure*** (*Figure 6-92*) describes the use of a stop by the start or end of a ***JourneyPatternTimingLink,*** or unordered stop reference in a FlexibleJourneyPattern***.*** It provides default values that will be inherited by the corresponding ***VehicleJourneyStopUsage*** elements of dependent vehicle journeys.

Both ***JourneyPatternStopUsage*** and ***VehicleJourneyStopUsage*** instances can be identified by a unique ***id*** attribute, and may also have a ***SequenceNumber*** attribute to indicate the preferred ordering of stops when presenting schedules in matrix timetable formats.

JourneyPatternStopUsage comprises a number of elements falling into two groups:

1. ***JourneyStopUsageGroup:*** Shared elements common to journey pattern stop usage elements, and to vehicle journey stop usage elements.
2. ***JourneyPatternStopUsageGroup:*** Elements specific to journey pattern stop usage elements.

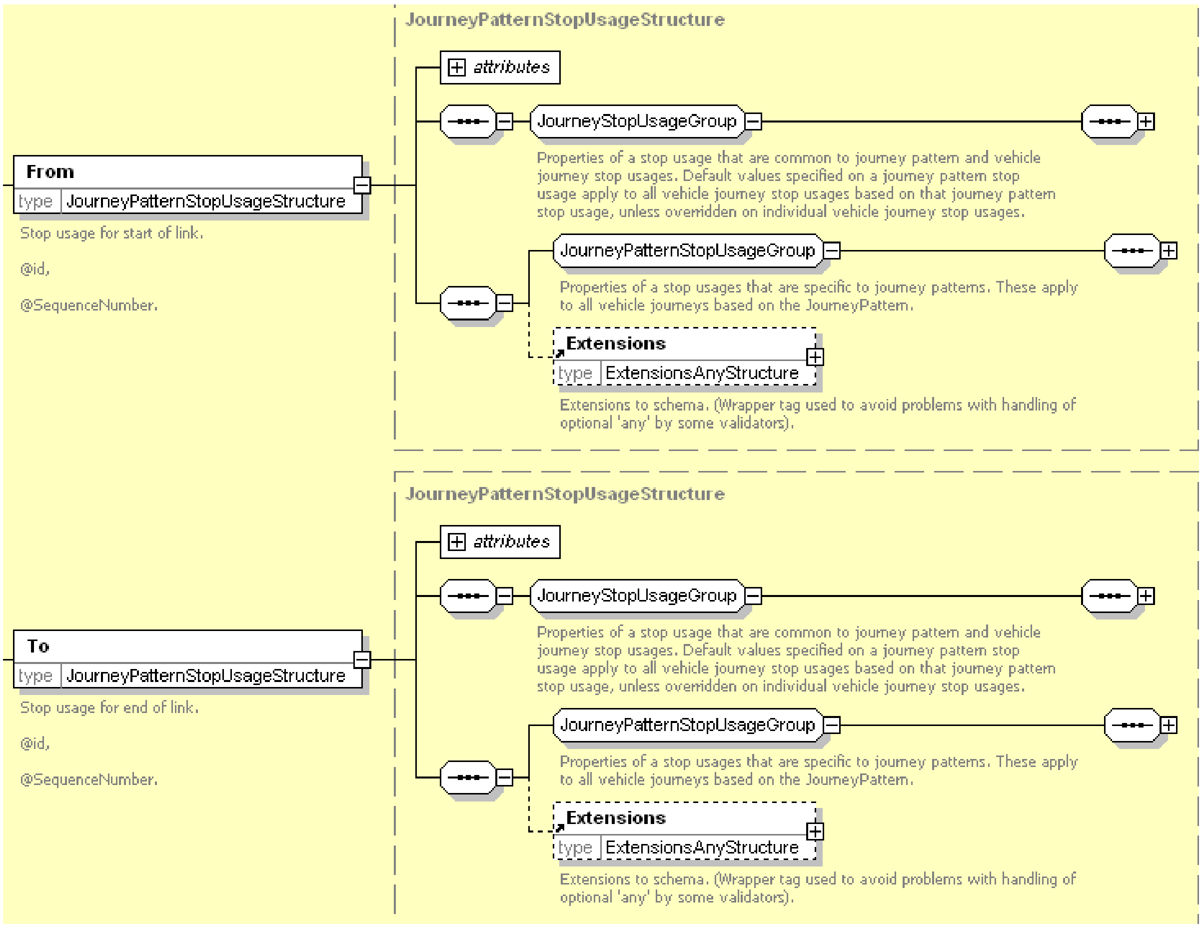


Figure 6-92 – JourneyPattern / JourneyPatternStopUsageStructure

6.7.8.1 JourneyPatternStopUsage / JourneyStopUsageGroup

The ***JourneyStopUsageGroup*** (*Figure 6-93*) holds elements that are common to both a ***JourneyPatternStopUsage*** and a ***VehicleJourneyStopUsage.*** Default values specified on a journey pattern stop usage apply to all vehicle journey stop usages based on that journey pattern stop usage, unless overridden on individual vehicle journey stop usages.

- ***WaitTime***: Time to wait at the referenced stop; the wait time is the part of the *Overall Wait Time* at the stop that has been ascribed to end of the link represented by the stop usage. When calculating departure times for a specific vehicle journey, the timing link ***WaitTime*** *values* from the respective stop usage ends of the incoming and outgoing links are added together to create the total wait time at the stop. If not specified, assume zero.
- ***Activity***: Activity undertaken by vehicle at stop. See *Table 6-48*. Defaults to *pick up and set down*.

Value	Description
pickUp	Pick up passengers.
setDown	Set down passengers.
pickUpAndSetDown	Pick up and set down passengers.
pass	Do not stop at stop.

Table 6-48 – Allowed Values for Activity

- ***DynamicDestinationDisplay***: Journey destination applicable to vehicle at referenced stop.
- ***Vias***: Names of intermediate points applicable to vehicle at referenced stop. If not specified on a VehicleJourney, value will be defaulted from JourneyPatternStopUsage. (+TXC v2.4)
  - **ViaName:** Name of an intermediate point on the journey to be shown as a via point.
  - ***None***: There are no Vias to be shown
- ***VariableStopAllocation***: In bus stations, bays may be allocated to a service variously on different days. This can be specified using the ***VariableStopAllocation*** element. See below.

- ***StopOnlyOnRequest***: Whether stop is only a request stop on this journey. Default false.
- ***Note***: Descriptive text note associated with stop.
- ***StopAccessibility***: Accessibility details for stop – applicable for specific visit of service (TXC V2.5).

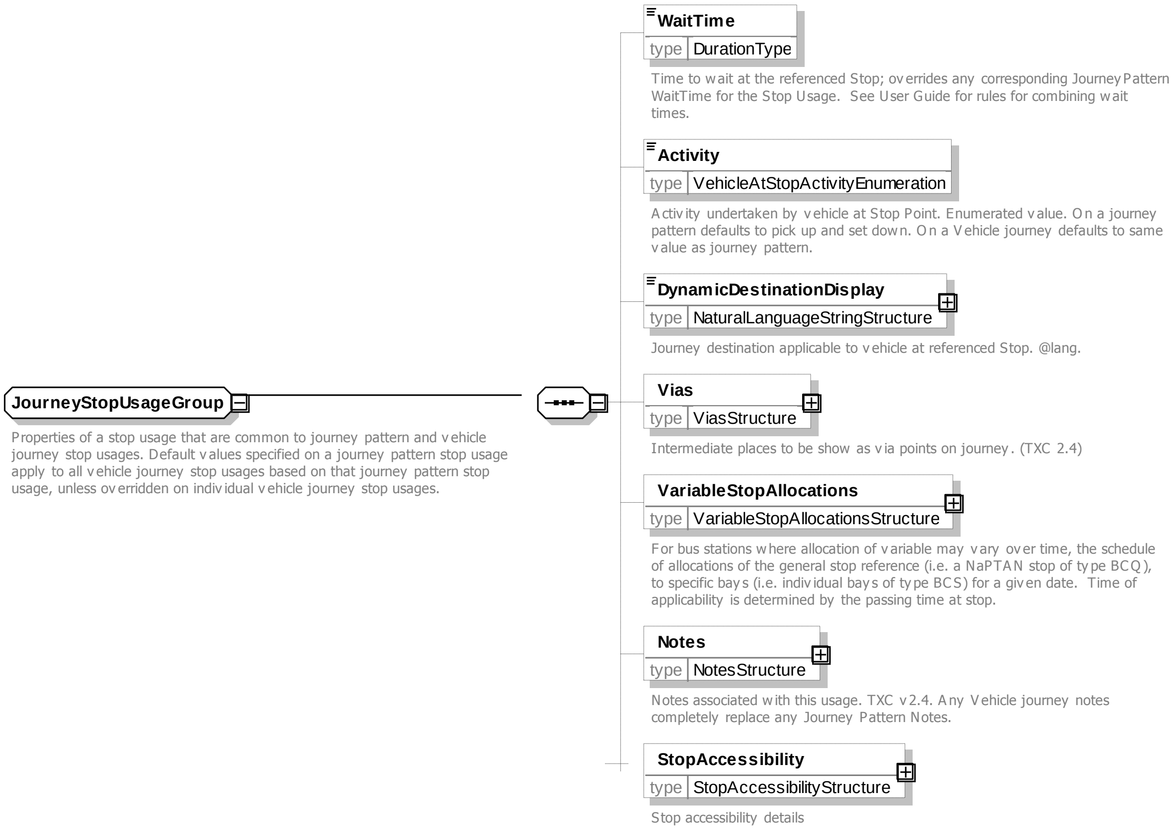


Figure 6-93 – JourneyPattern / JourneyStopUsageGroup

6.7.8.2 JourneyPatternStopUsage / JourneyPatternStopUsageGroup

The ***JourneyPatternStopUsageGroup*** (*Figure 6-94*) holds information specific to a ***JourneyPatternStopUsage:***

- ***StopPointRef***: *NaPTAN* Stop at which timing link starts or ends.
- ***TimingStatus***: Classification of the role of the stop as a timing point used by the journey pattern. See *Table 6-49*. Overrides the classification defined by the stop in *NaPTAN*.

Value	Long Value	Description
PTP	*principalTimingPoint*	Principal and time info point.
TIP	*timeInfoPoint*	Time Info Point.
OTH	*otherPoint*	Other Bus Stop.

Table 6-49 – Allowed Values for TimingStatus

- ***FareStageNumber***: The fare stage number for the referenced stop. A fare stage number should be specified if the fare stage is different from that on the previous link.
- ***FareStage***: Whether a fare stage is encountered while traversing the end of the timing link. This should correspond to the value implied by the ***FareStageNumber***. If the two are in conflict, then the ***FareStageNumber*** will be assumed correct.

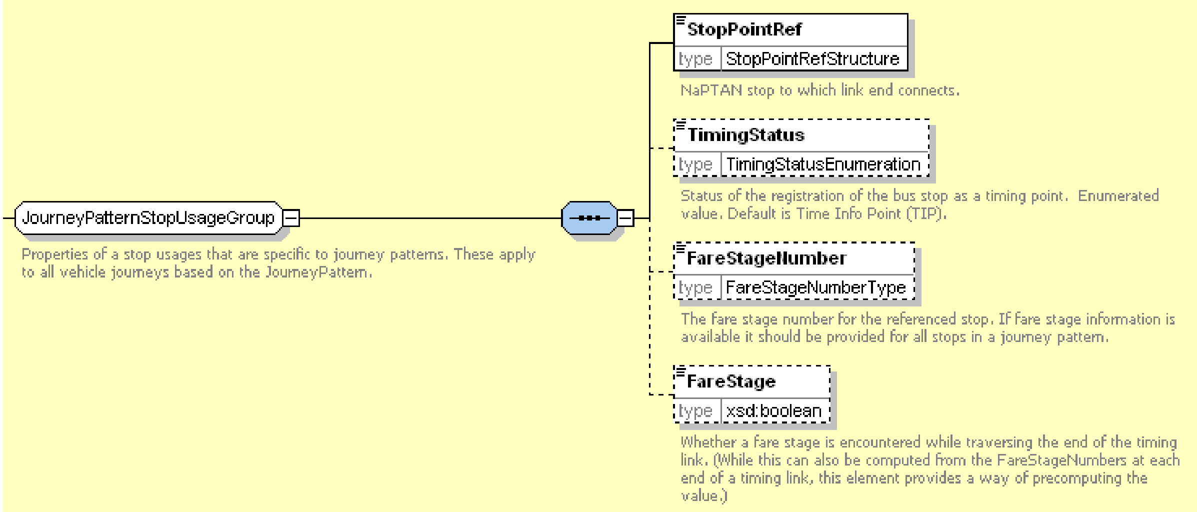


Figure 6-94 – JourneyPattern / JourneyPatternStopUsageGroup

6.7.8.3 VariableStopAllocations Element

The ***VariableStopAllocations*** element (*Figure 6-95*) describes the variable allocation of bays in a bus station. It can be used to assign to assign specific bays for a service on specific dates. It comprises zero or more ***VariableStopAllocation*** elements, each specifying an individual allocation on a given date. The time of allocation is the passing time of the service at the stop. Normally the assigning stop will be of stop type '*BCQ' (Bus / Coach Station Variable Bay)*, the assigned stops of type '*BCT' (Bus / Coach Station Bay)*.

- ***DateRange***: A collection of one or more open-ended date ranges, and any number of date exceptions.
  - ***StartDate***: The (inclusive) start date. If omitted, the range start is open-ended, that is, it should be interpreted as "since the beginning of the service validity period".
  - ***EndDate***: The (inclusive) end date. If omitted, the range end is open-ended, that is, it should be interpreted as "until end of the service validity period" (which may be indefinite).
- ***VariableStopPoint***: Bay or bays to which service is allocated for the specified date (and time of the service). Normally will be a *NaPTAN* stop of type '*BCT (Bus / Coach Station Bay)'*. If more than one stop is specified, then bays are considered to be a pool that can be used on a first come first serve basis.
  - ***StopPointRef***: *NaPTAN* Identifier of a ***StopPoint***.
- ***DefaultStopAllocation***: Bay or pool of bays to use if no date-specific ***VariableStopAllocation*** is applicable for a given date.

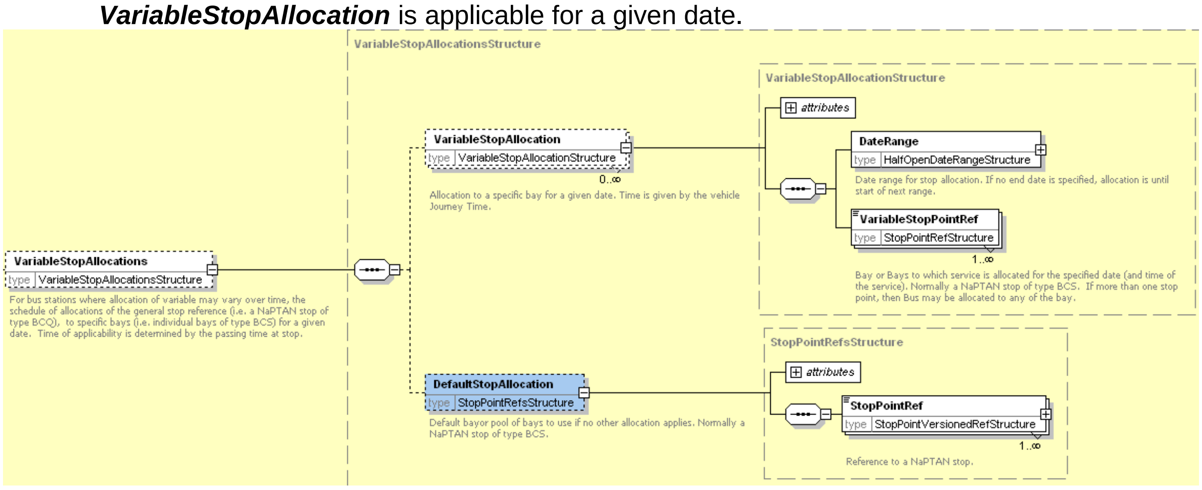


Figure 6-95 – JourneyPattern / VariableStopAllocation Element

6.7.8.4 StopAccessibility Element (+TXC v2.5)

The ***StopAccessibility*** element (Figure 6-96) specifies the accessibility of the visit to the stop for mobility impaired users. This can be used to override the Journey level defaults if there are specific considerations for a particular stop. For example, it may be that a Vehicle is normally accessible using equipment but that the hoist cannot be used at certain stops. It comprises an overall assessment and a number of detailed criteria. It comprises:

- A ***MobilityImpairedAccess:*** Overall assessment of the stop for accessibility. This can be used for example to indicate accessible and inaccessible stops on maps and in journey planners. See Limitation Status (Table 6-50) for allowed values. For a topologically simple stop such as an on-street bus stop, this will typically be the same as the ***WheelchairAccess*** status. For complex stops such as metro and rail stations it requires an overall judgement based on the accessibility of individual platforms. For example a station which requires the use of a flight of steps to reach the main platform would be considered inaccessible.

Specific assessments:

- ***WheelchairAccess:*** Whether stop is accessible to wheelchair users. See Limitation Status (Table 6-50) for allowed values. Normally if there is Step free access there will be wheelchair access. However wheelchair access may additional require assistance, use of a boarding ramp etc.
- ***StepFreeAccess:*** Whether stop is accessible without the use of steps. Normally only relevant for off-street stops such as Station platforms. See Limitation Status (Table 6-50) for allowed values.
- ***EscalatorFreeAccess:*** Whether stop is accessible without the use of escalator. Normally only relevant for off-street stops such as Station platforms. See Limitation Status (Table 6-50) for allowed values.
- ***LiftFreeAccess:*** Whether stop is accessible without the use of lifts. Normally only relevant for off-street stops such as Station platforms. See Limitation Status (Table 6-50) for allowed values. Lift free access may be of concern to sufferers from claustrophobia, autism and other conditions.

Limitation Status (Table 6-50).shows the allowed values for accessibility assessments. Note that a value of unknown should be used if the accessibility is not known.

Value	Description
true	Stop is considered accessible according to criteria.
false	Stop is not considered accessible according to criteria.
partial	Stop is partial accessible according to criteria: some areas are not accessible.
unknown	The accessibility of the stop according to the criteria a not known.

Table 6-50 – Allowed Values for LimitationStatus

The Limitation Status includes an “unknown” value which can be used when the accessibility status is not known. It is reasonable to assume that Air, Bus and Coach Stops will usually be accessible even if a value is not specified. See Table 6-51.

Mode of stop	Value to assume if unspecified
Air	True
Rail	Unknown
Metro	Unknown
Ferry	Unknown
Tram	Unknown
Bus	True
Coach	True

Table 6-51 – Accessibility defaults by mode

Assistance values:

- ***AccessVehicle:*** Details on accessibility for wheelchair users. See below
- ***AssistanceAvailability:*** Whether there is an assistance service available for disabled users and if so how it should be booked. See Table 6-8 earlier for allowed values.
- ***AssistanceTimes:*** Times when assistance is available.

- *DayType:* Type of day (Monday to Sunday) and ***Timeband.*** See below.
- ***OperatorRef:*** Identifier of operator who provides service. This can be used to integrate booking details and other information.
- ***InfoUrl:*** Public URL with information about accessibility at the stop.

Further details

- ***ServicesAtStopAreNormallyAccessible:*** Whether services at the stop are normally accessible, for example the vehicle type has low floor, a wheelchair hoist, etc. This is a default value for indicative guidance only. It may be that specific services are not accessible. See Limitation Status (Table 6-50) for allowed values.
- ***Note:*** Any comment about the accessibility at the stop for the journey.

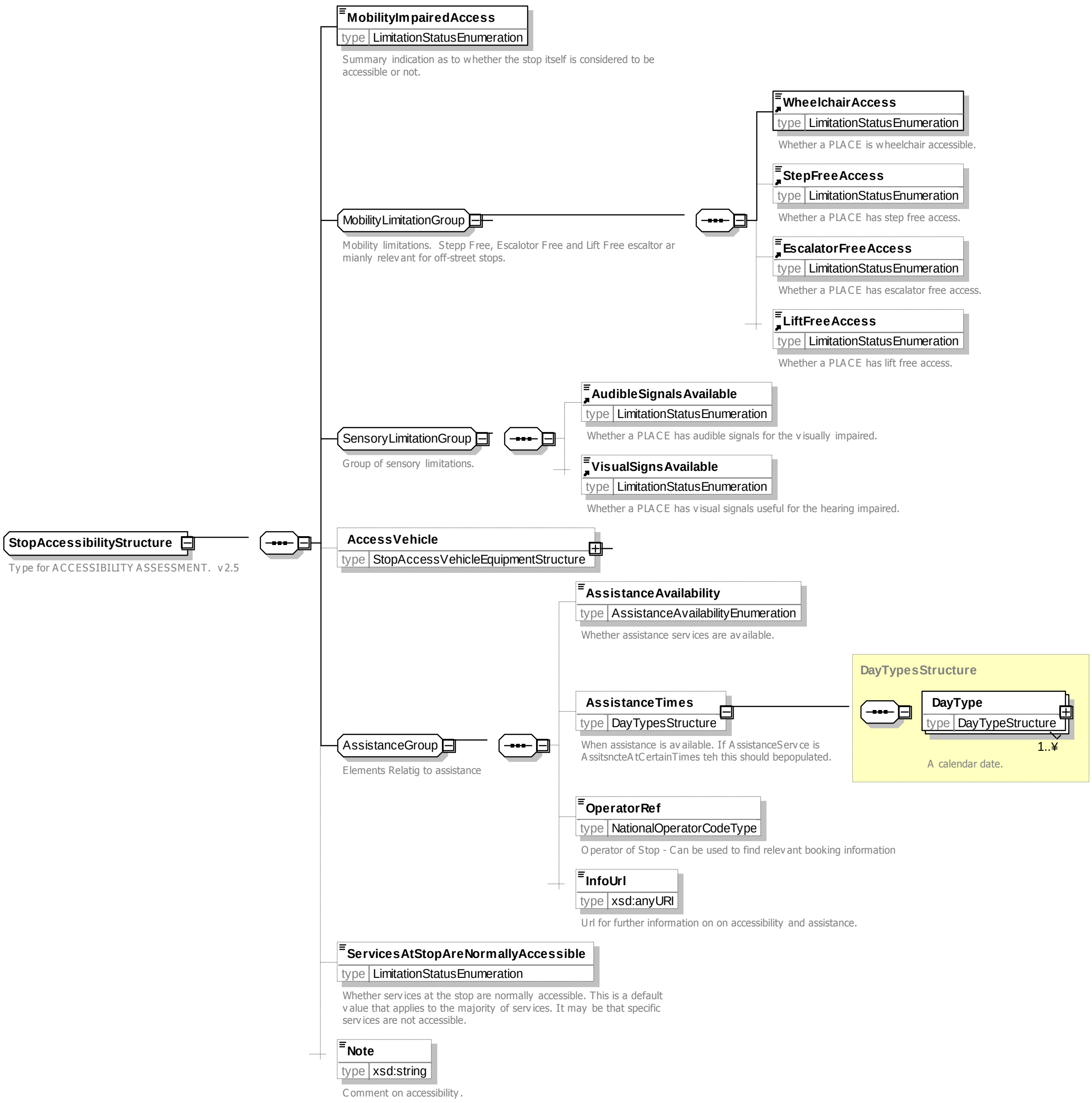


Figure 6-96 – StopAccessibility Element

6.7.9 JourneyPatternInterchange Element

The ***JourneyPatternInterchange*** element (*Figure 6-97*) describes an interchange connecting two ***JourneyPatterns***. Each interchange can be identified by a unique ***id*** attribute, and comprises a number of elements, falling into two groups:

1. ***CommonInterchangeGroup:*** Shared elements common to journey pattern interchanges and vehicle journey interchanges. See below.
2. ***JourneyPatternInterchangeGroup:*** Elements specific to vehicle journey interchanges. See below.

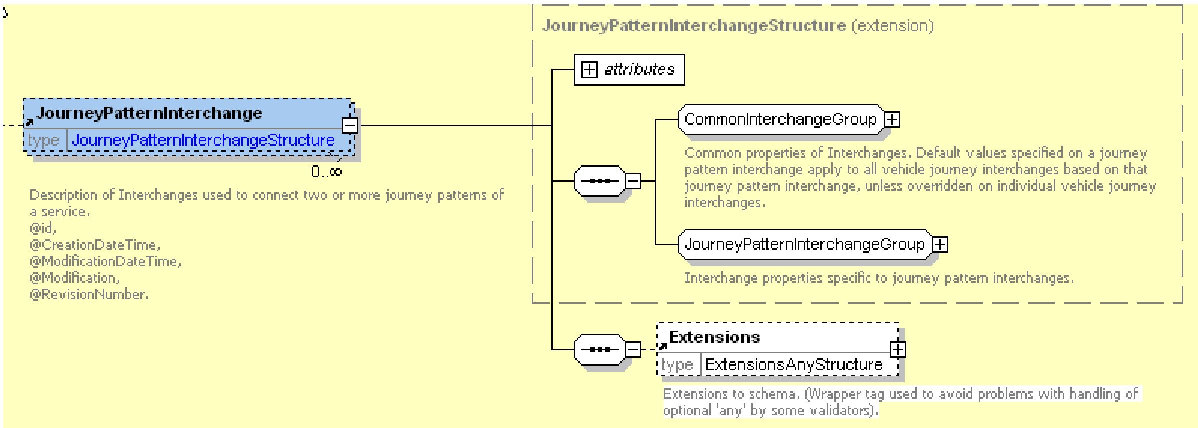


Figure 6-97 – JourneyPatternInterchange Element

6.7.9.1 JourneyPatternInterchange / CommonInterchangeGroup

The ***CommonInterchangeGroup*** (*Figure 6-98*) holds information that is common to both a ***JourneyPatternInterchange*** and a ***VehicleJourneyInterchange.***

- ***MinInterchangeTime***: Minimum time to allow for changing services at the interchange.
- ***MaxInterchangeTime***: Maximum time that connecting service will wait at the interchange.
- ***TransferMode***: Method of transport used to make transfer between inbound and outbound journeys at the interchange. See *Table 6-52*.

Value	Description
walk	Walk transfer.
bus	Bus transfer.
train	Train transfer.
tram	Tram transfer.
metro	Metro transfer.
coach	Coach transfer.
ferry	Ferry transfer.
air	Air transfer.
taxi	Taxi transfer.
cycle	Cycle transfer.
movingWalkway	Moving Walkway transfer.

Table 6-52 – Allowed Values for TransferMode

- ***ValidityPeriod***: Period when the interchange is valid.
  - ***StartDate***: Inclusive date of start of validity period.
  - ***EndDate***: Inclusive date of end of validity period.
- ***StoppingArrangements***: Text description of stopping arrangements for the interchange.
- ***InterchangeActivity:*** Activity taking place between incoming and outgoing ***VehicleJourney*** instances at an interchange. See *Table 6-53.*
- ***InterchangeInfoGroup***: Additional information about the nature of the interchange**.** See below.

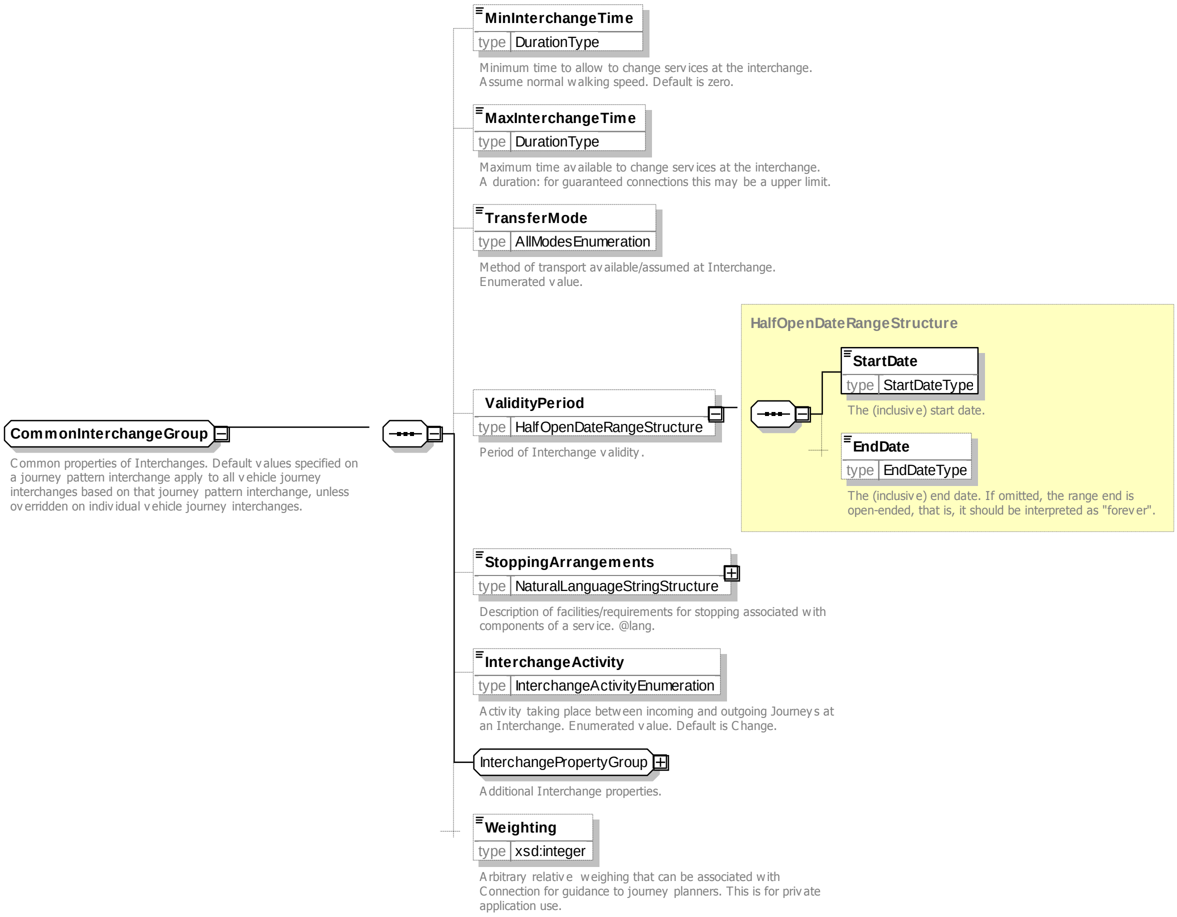


Figure 6-98 – CommonInterchangeGroup

6.7.9.2 JourneyPatternInterchange / InterchangeInfoGroup

The ***InterchangeInfoGroup*** (*Figure 6-99*) holds additional information about the nature of the interchange***.***

- ***CrossBorder***: Whether the connection crosses a border.
- ***GuaranteedConnection:*** Whether the connection is guaranteed.
- ***ChangeLineNumber***: Whether the service changes number at the connection.

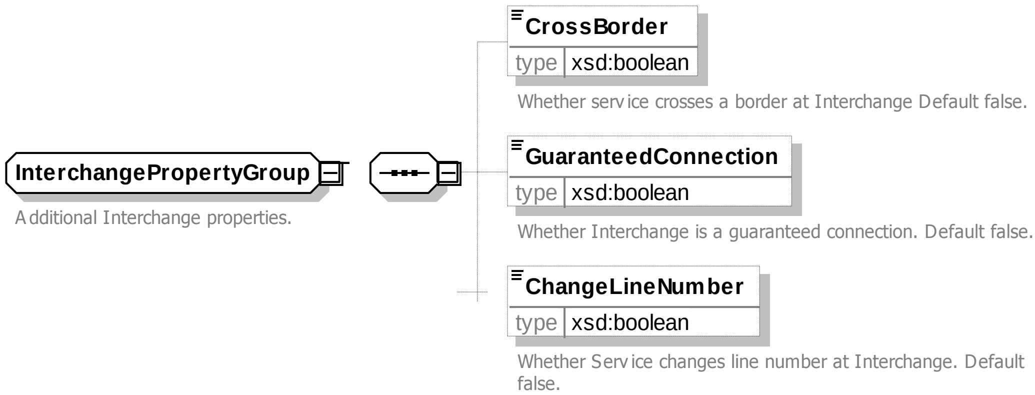


Figure 6-99 – JourneyPatternInterchange / InterchangeInfoGroup

6.7.9.3 JourneyPatternInterchange / JourneyPatternInterchangeGroup

The ***JourneyPatternInterchangeGroup*** holds elements that are specific to a ***JourneyPatternInterchange,*** and describe the connection between two journeys.

- ***Inbound***

- ○ ***JourneyPatternRef***: Incoming ***JourneyPattern*** that connects to the interchange.
- ○ ***StopUsageRef***: Reference to the ***JourneyPatternStopUsage*** of the ***JourneyPatternTimingLink*** that connects inbound ***JourneyPattern*** to the interchange.

- ***Outbound***
  - ○ ***JourneyPatternRef***: On-going ***JourneyPattern*** that connects from the interchange.
  - ○ ***StopUsageRef***: Reference to the ***JourneyPatternStopUsage*** of the ***JourneyPatternTimingLink*** that connects the outbound ***JourneyPattern*** to the interchange.

Value	Description
change	Service changes at interchange
join	Service joins at interchange.
split	Service splits at interchange.
through	Through journey.

Table 6-53 – Allowed Values for InterchangeActivity

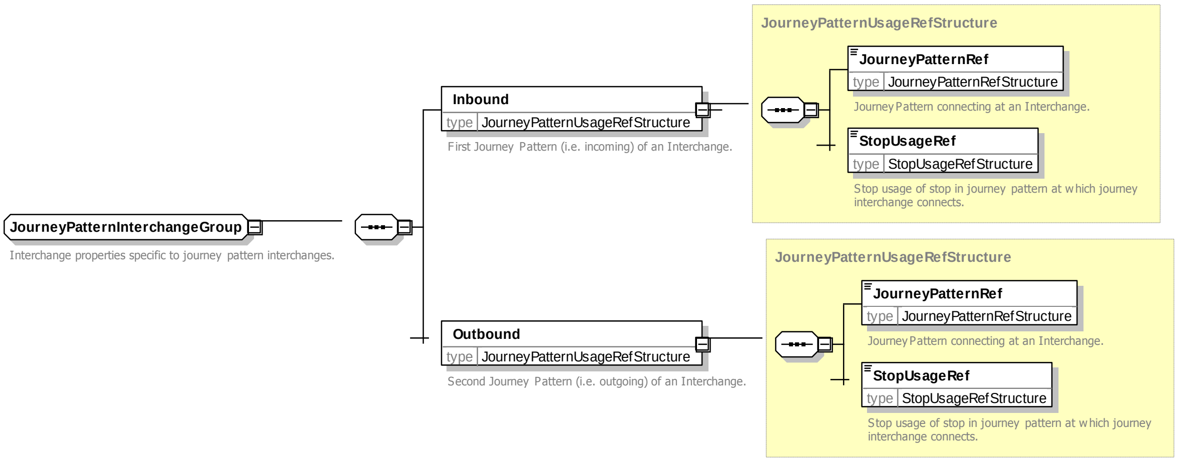


Figure 6-100 – JourneyPatternInterchange / JourneyPatternInterchangeGroup

6.8 VehicleJourney Element

A ***VehicleJourney*** (*Figure 6-101*) describes a specific journey of a vehicle following a fixed ***JourneyPattern*** of a ***StandardService***. The ***JourneyPattern*** comprises one or more ***VehicleJourneyTimingLink*** elements: the order of links represents the order in which they are traversed. A ***VehicleJourney*** comprises a number of elements; the elements fall into three groups:

1. ***CommonJourneyGroup:*** Shared elements common to journey patterns and vehicle journeys. See ***JourneyPattern*** / ***CommonJourneyGroup*** earlier. Allows individual properties to be overridden on a vehicle journey: if not specified the property from the journey pattern will be used. The facility to override information from the JourneyPattern in the VehicleJourney should be used sparingly. Information that is shared many times with other journeys following the same journey pattern should be added to the journey pattern and not be repeated unnecessarily.
2. ***VehicleJourneyGroup:*** Elements specific to vehicle journeys, both fixed and flexible.
3. ***StandardVehicleJourneyGroup:*** Elements specific to fixed route vehicle journeys.

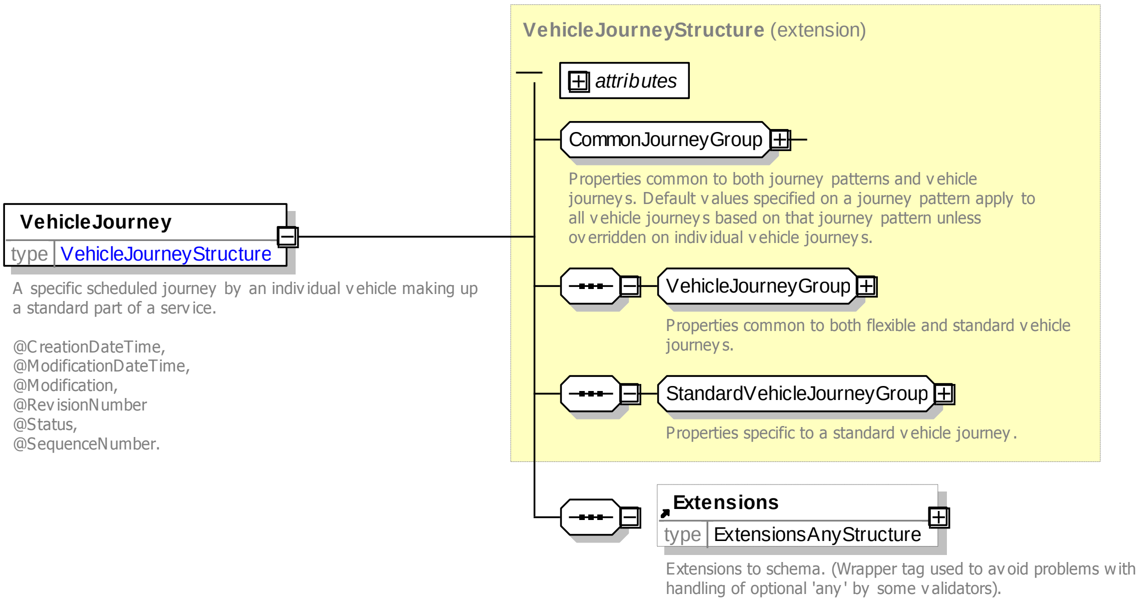


Figure 6-101 – VehicleJourney Element

6.8.1 VehicleJourney / VehicleJourneyGroup

The ***VehicleJourneyGroup*** (*Figure 6-102*): holds elements that are common to both fixed and flexible types of ***VehicleJourney.***

- ***VehicleJourneyCode***: A unique code that can be used to identify the ***VehicleJourney***.
- ***ServiceRef***: The ***Service*** to which the ***VehicleJourney*** belongs.
- ***LineRef***: The ***Service*** / ***Line*** that the ***VehicleJourney*** serves.
- Referenced Journey pattern. One of the following:
    - ***JourneyPatternRef***: The ***JourneyPattern*** over which the ***VehicleJourney*** runs. Route, timing links and other properties will be derived from the specified journey pattern.
    - ***VehicleJourneyRef***: Reuse the ***VehicleJourneyTimingLink*** elements of the referenced ***VehicleJourney***, and follow its ***JourneyPattern.*** If a ***VehicleJourneyRef*** is specified, then any ***VehicleJourneyTimingLink*** instances of the dependent ***VehicleJourney*** will be ignored.
- ***StartDeadRun:*** Initial "dead run" for positioning the vehicle before it traverses its timing links. See below.
- ***EndDeadRun:*** Final "dead run" link for positioning the vehicle after it traverses its timing links. See below.

- ***VehicleJourneyInterchange:*** Interchanges where the vehicle journey connects with another vehicle journey. See later.
- ***Note:*** Any additional notes on the ***VehicleJourney***. See below.

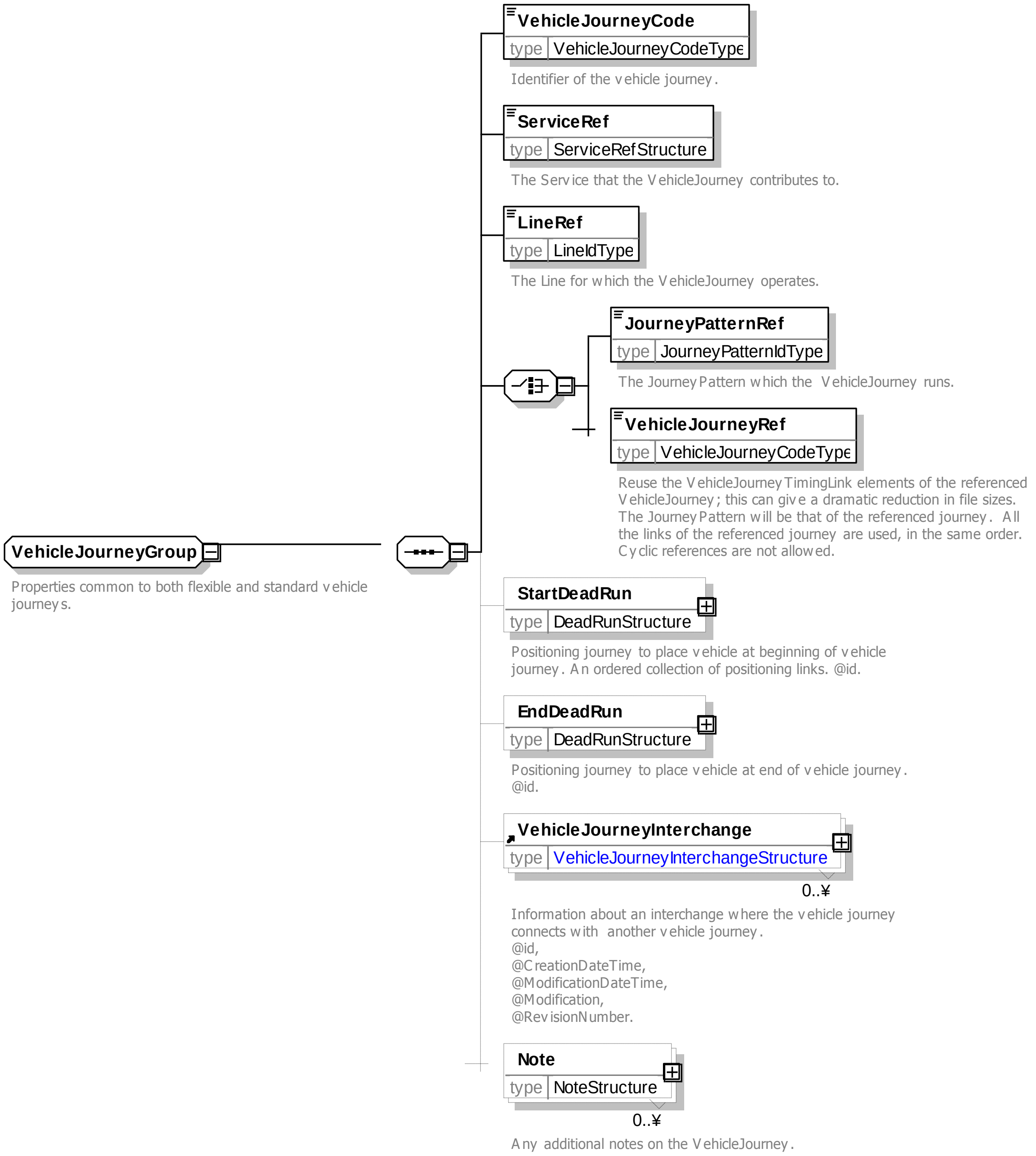


Figure 6-102 – VehicleJourney / VehicleJourneyGroup

6.8.2 VehicleJourney / StandardVehicleJourneyGroup

The ***StandardVehicleJourneyGroup*** (*Figure 6-103*) holds elements that are specific to fixed ***VehicleJourney*** instances:

- ***DepartureTime:*** Time of departure from origin stop of the ***VehicleJourney***.
- ***DepartureDayShift:*** Whether the ***DepartureTime*** is to be shown as the same day (0 or omit,) the next day (+1) or the Previous day. (-1) This allows Journeys that start or end after midnight to be included in a particular bed, do for example a journey that runs in at 0-0:05 that is part of the Monday to Friday service can be shown as a as expressed by the profile (+TXC v2.4). See earlier discussion of the effect of

- ***Frequency***: Describes service frequency for frequency based services. See below.
- ***VehicleJourneyTimingLink:*** An ordered collection of timing links making up the ***VehicleJourney***. See ***VehicleJourneyTimingLink*** later.

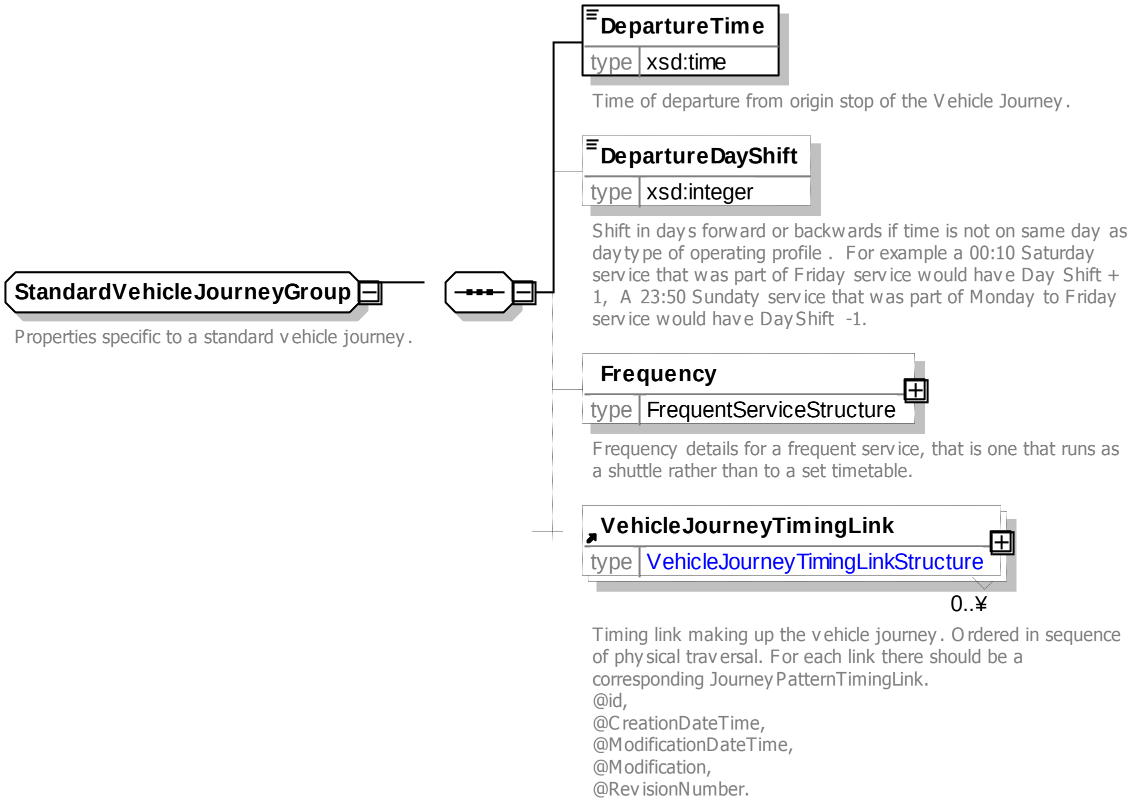


Figure 6-103 – VehicleJourney / StandardVehicleJourneyGroup

6.8.3 Common VehicleJourney Subelements

6.8.3.1 VehicleJourney / DeadRun Element

A ***DeadRun*** (*Figure 6-104*) models a ***StartDeadRun*** or ***EndDeadRun,*** that is, a positioning run at the start or end of a journey; it is used to place a vehicle in position to start the service, or to retrieve it at the end of the journey.
A DeadRun can be used to track the vehicle in a real time system to give better predictions for the part of the journey that is in public service, and so should be provided along with the run times that relate to the running of the vehicle during the positioning journey (i.e. as a ***JourneyPatternTiming-Link*** with ***RunTimes)***. The dead run should not be treated as an out of service part of the journey in order to hide that part of the journey from the public timetables unless the vehicle actually operates out of service to the stated times.
A ***DeadRun*** comprises:

- ***PositioningLink:*** One or more links describing how the vehicle travels to or from the route. See below.
- ***ShortWorking:*** If the dead run intercepts the journey pattern at a point, identifies the start or end point on the journey pattern at which the interception happens. May be used even if no positioning link is specified.
    - ***JourneyPatternTimingLinkRef:*** Link at which journey starts or finishes.

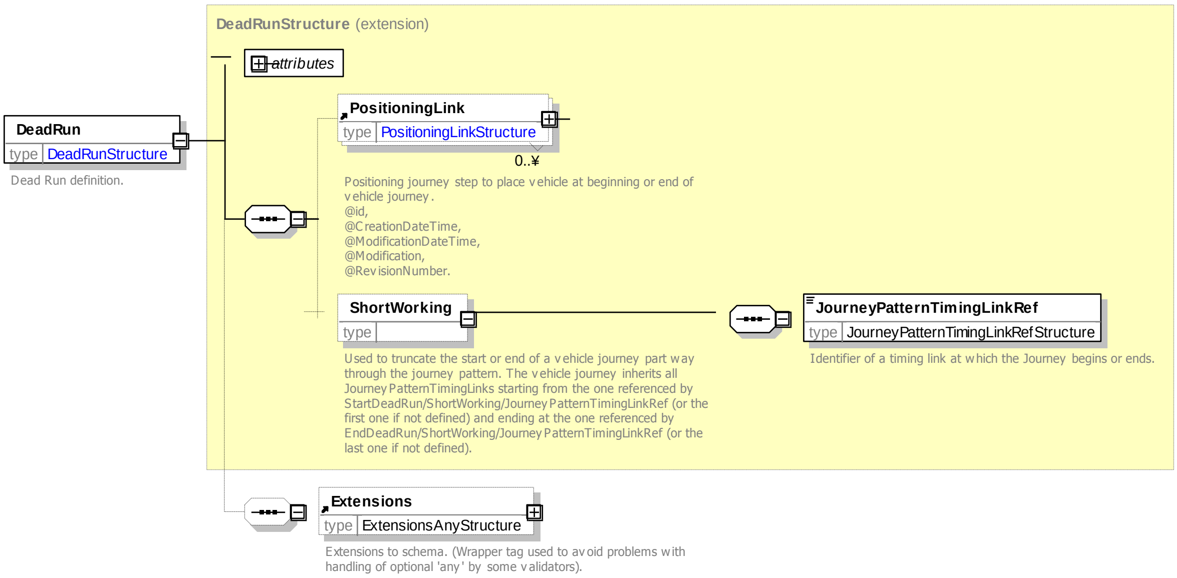


Figure 6-104 – VehicleJourney / DeadRun Element

6.8.3.2 VehicleJourney / PositioningLink Element

A ***PositioningLink*** (*Figure 6-105*) models a step of a ***DeadRun.*** It comprises:

- ***RunTime:*** Time taken to traverse link.
- ***From:*** From point, a stop, garage, or location. See ***PositioningLinkUsage*** below.
- ***To:*** To point; also a stop, garage, or location. See ***PositioningLinkUsage*** below.
- ***DutyCrewRef:*** Identifier of crew manning vehicle over link. ***(+TXC v2.4)***
- ***Track:*** Path taken by vehicle when traversing the positioning link. See ***RouteLink / Track*** element earlier.

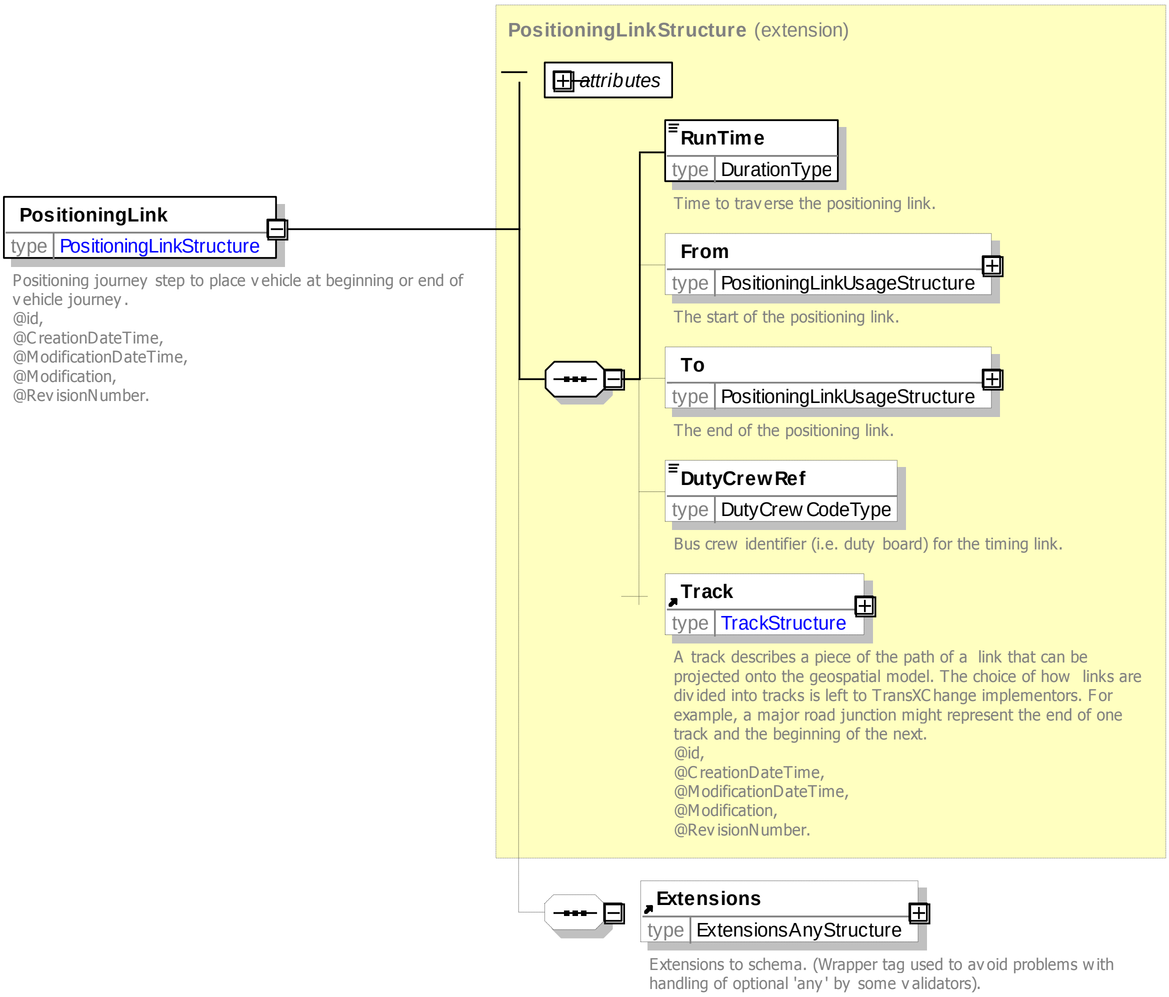


Figure 6-105 – DeadRun / PositioningLink Element

6.8.3.3 VehicleJourney / PositioningLink / PositioningStopUsageStructure

A ***PositioningLinkUsage*** (*Figure 6-106*) models one end of a ***PositioningLink.*** It comprises one of the following:

- ***StopPointRef:*** A *NaPTAN* stop point. Usually on the journey pattern, but can be completely arbitrary – e.g. a stop on another route from which the bus is coming.
- ***GarageRef:*** A ***Garage*** defined for the operator of the ***Service***. to which vehicle journey belongs
- ***LayoverPointRef:*** A ***LayoverPoint*** defined for the ***JourneyPattern***.
- ***Location:*** An arbitrary location specified by spatial coordinates.

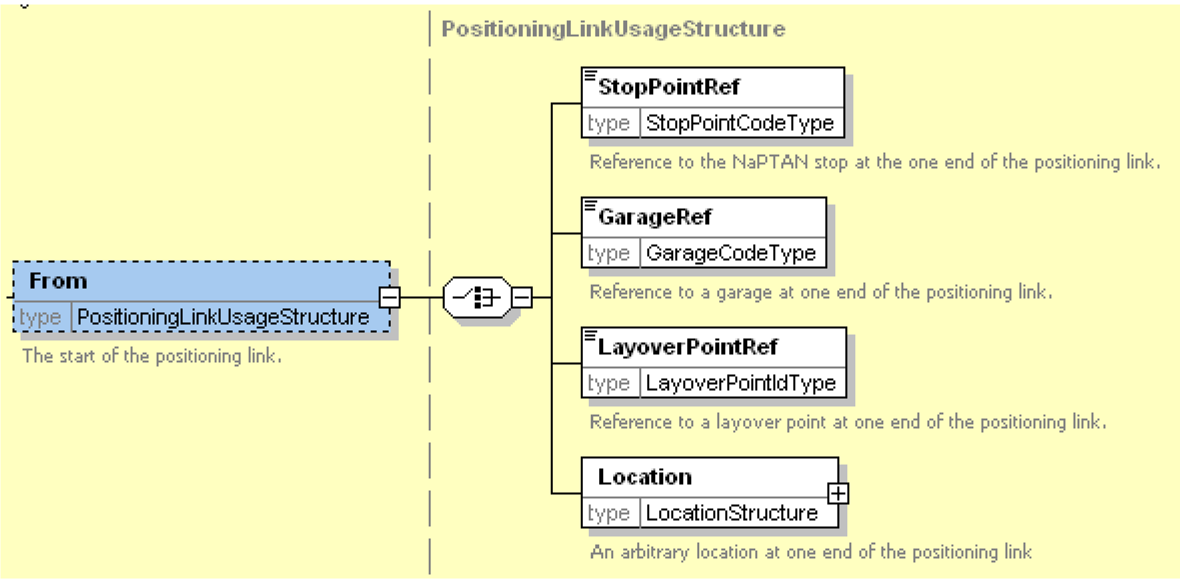


Figure 6-106 – DeadRun / PositioningLinkUsageStructure

6.8.3.4 VehicleJourney / Frequency Element

Frequency (*Figure 6-107*) gives details about a frequency based service, that is, one that runs as a shuttle rather than to a set timetable.

- ***EndTime***: Describes when the frequency based period ends.

The frequency can be specified in one of two ways:

  - ***Interval***: Describes the expected frequency of a service in quantitative terms as an interval. See 6.8.3.5 below.

- ***Minutes past the hour:*** Describes the expected frequency of a service in quantitative terms. Comprises:
  - ***Minutes***: One or more times past the hour.
- ***FrequentService:*** Formally declares the journey to be a frequent service, with an interval of at least once every 10 minutes. A minimum frequency should be specified.

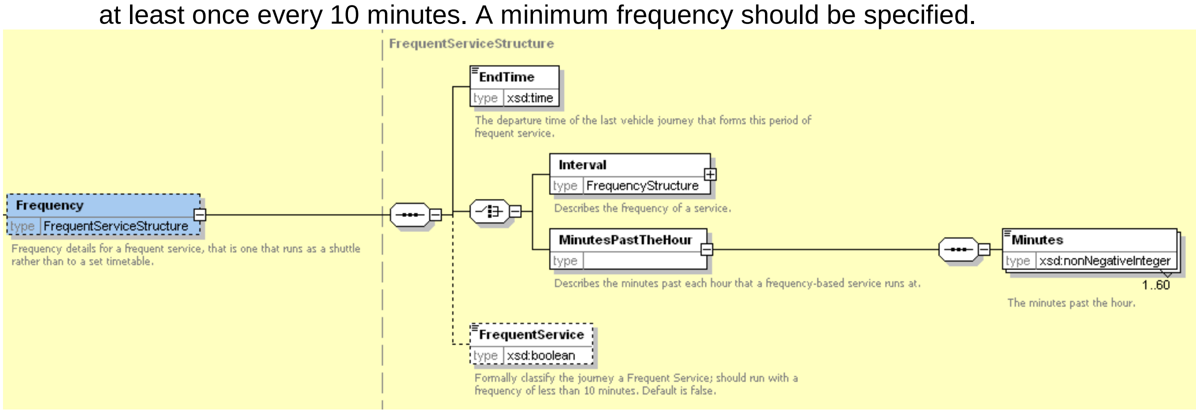


Figure 6-107 – VehicleJourney / Frequency Element

6.8.3.5 Frequency Interval

Interval: element describes the expected frequency of a service in quantitative terms as an interval, that can be used to describe the service periodicity such as "*Every ten minutes*". Comprises:

- ***ScheduledFrequency:*** The scheduled time gap between departures.
- ***MinimumFrequency***: The minimum time gap between departures.
- ***MaximumFrequency***: The maximum time gap between.
- ***Description***: Override description text for *Frequency*.

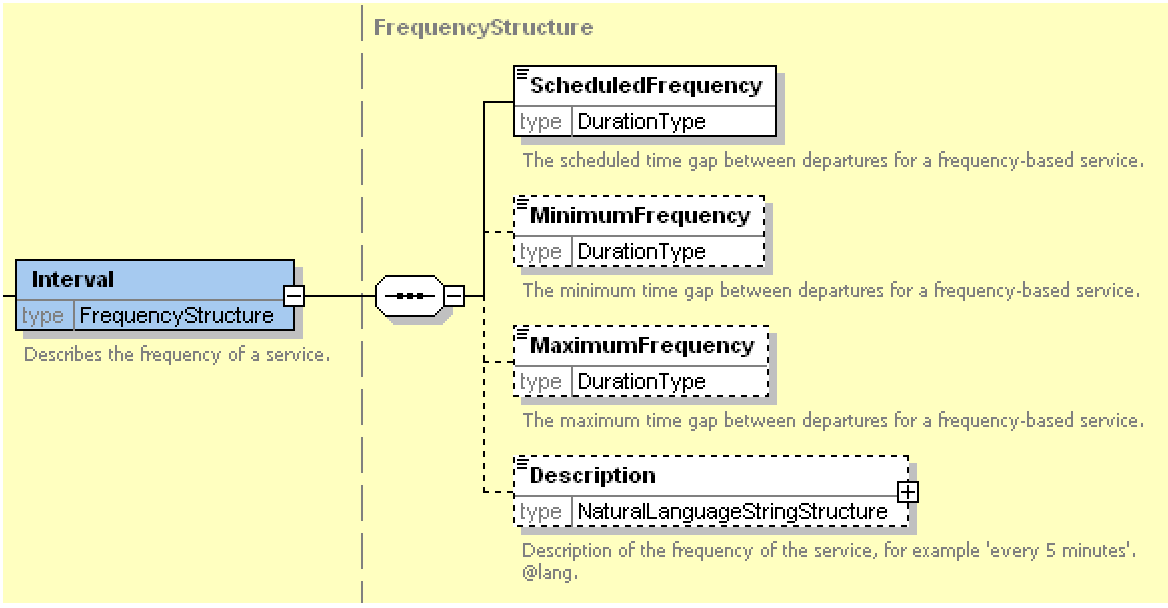


Figure 6-108 – Frequency Interval Element

6.8.4 VehicleJourneyTimingLink Element

A ***VehicleJourneyTimingLink*** (*Figure 6-109*) models the link between two stops of a vehicle journey. Each ***VehicleJourneyTimingLink*** can be identified by a unique ***id*** attribute, and comprises a number of elements. The elements fall into two groups:

1. ***CommonTimingLinkGroup:*** Shared elements common to journey pattern timing links and vehicle journey timing links. See ***JourneyPatternTimingLink / CommonTimingLinkGroup*** earlier.
2. ***VehicleJourneyTimingLinkGroup:*** Elements specific to vehicle journey timing links.

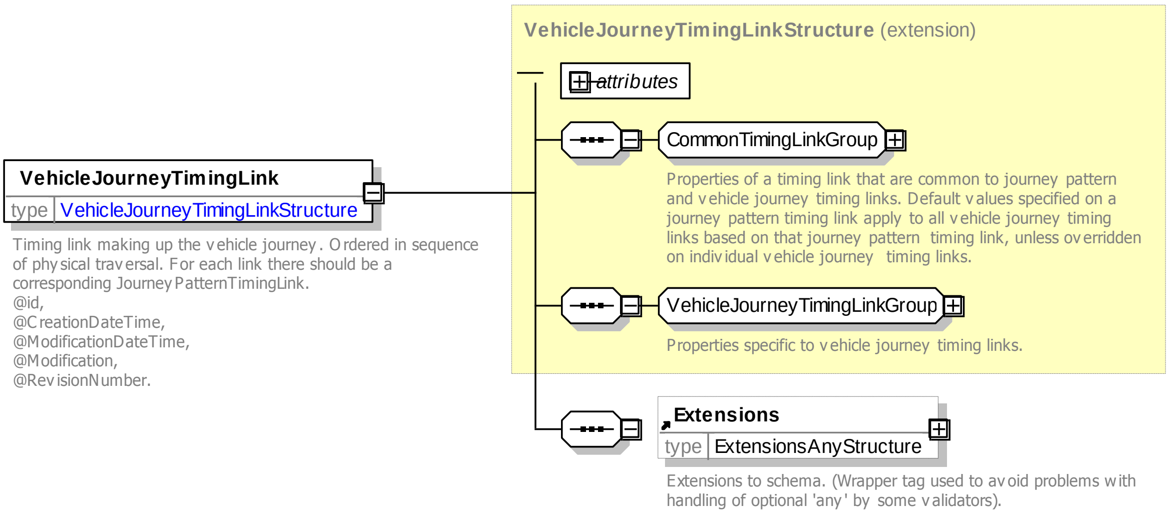


Figure 6-109 – VehicleJourneyTimingLink Element

6.8.4.1 VehicleJourneyTimingLink / VehicleJourneyTimingLinkGroup

The ***VehicleJourneyTimingLinkGroup*** (*Figure 6-110*) holds information is specific to a ***VehicleJourneyTimingLink:***

- ***ParentVehicleJourneyRef:*** Optional reference to parent ***VehicleJourney***. Normally not stated as given by containing context, but may be specified when using the link as a stand-alone artefact. If already given by context, this value is ignored**.** (+TXC v2.4).

- ***JourneyPatternTimingLinkRef:*** Reference to a ***JourneyPatternTimingLink*** onto which timing link projects, and which defines the origin and destination points of the link. See ***JourneyPatternTimingLink*** earlier.
- ***RunTime:*** Time taken to traverse link. Defaults to value specified for ***JourneyPatternTimingLink***.
- ***From:*** Usage details of from stop, specified by a ***VehicleJourneyStopUsage*** element. This projects onto the ***From / JourneyPatternStopUsage*** of the corresponding ***JourneyPatternTimingLink***.
- ***To:*** Usage details of from stop, specified by a ***VehicleJourneyStopUsage*** element. This projects onto the ***To / JourneyPatternStopUsage*** of the corresponding ***JourneyPatternTimingLink.***

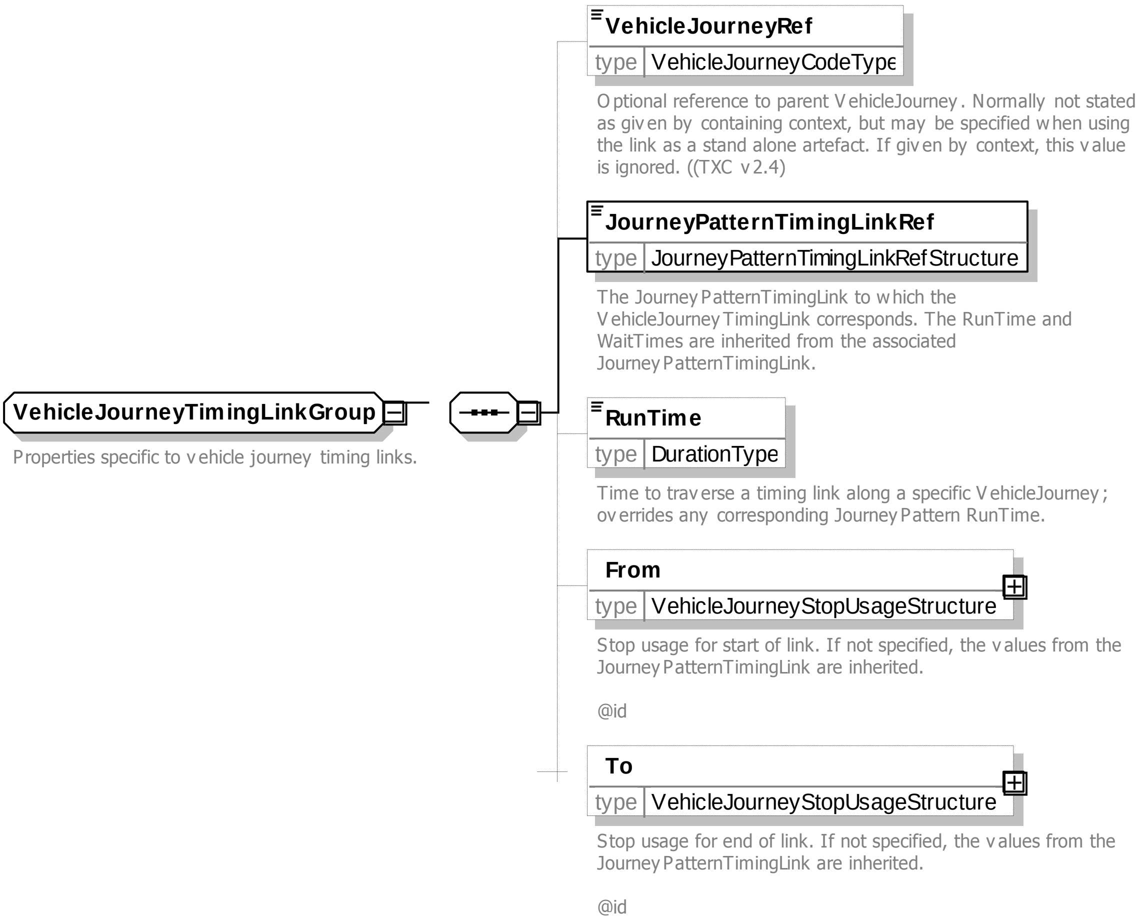


Figure 6-110 – VehicleJourneyTimingLinkGroup

6.8.5 VehicleJourneyTimingLink / VehicleJourneyStopUsage Element

The ***VehicleJourneyStopUsageStructure*** (*Figure 6-94*) describes the use of a stop by the start or end of a ***VehicleJourneyTimingLink.*** The ***VehicleJourneyStopUsage*** can be identified by a unique ***id*** attribute, and comprises:

- A ***JourneyStopUsageGroup***: see ***JourneyPatternStopUsage*** earlier. Any values specified override the values specified for the underlying journey pattern.
- A Frequency ***Interval***. See 6.8.3.5. If a value is specified, it indicates this is a Partial Frequency Based service and that the stop passing time should be presented as a frequency (e.g. "*Every five minutes*") rather than an absolute time (e.g. "*10:23*"). Note that an absolute time will still be computed in case absolute times are needed for any subsequent stops. See discussion of computing of passing times earlier. (+TXC v2.4). An override Interval will normally only be used only on the departure end of a timing link, that is for the ***From*** / ***Stop UsageGroup***.

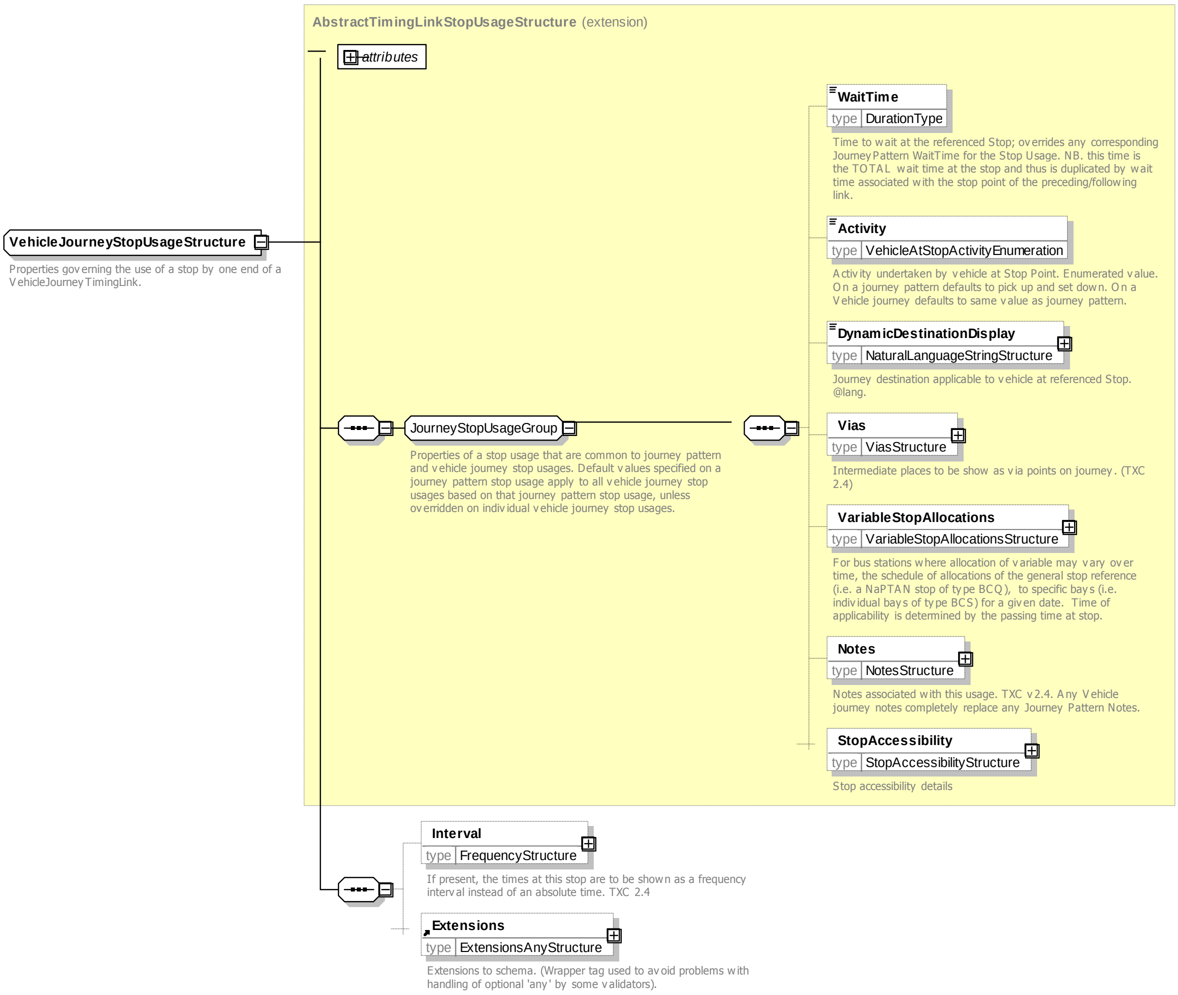


Figure 6-111 – VehicleJourneyStopUsage Element

6.8.6 VehicleJourney / VehicleJourneyInterchange Element

The ***VehicleJourneyInterchange*** element (*Figure 6-112*) records information about an interchange at which the vehicle journey connects with another vehicle journey. Each interchange can be identified by a unique ***id*** attribute, and comprises a number of elements, falling into two groups:

1. ***CommonInterchangeGroup:*** Shared elements common to journey pattern interchange and vehicle journey interchange elements. See ***JourneyPatternInterchange*** / ***CommonInterchangeGroup*** element earlier.
2. ***VehicleJourneyInterchangeGroup:*** Elements specific to vehicle journey interchange elements. See below.

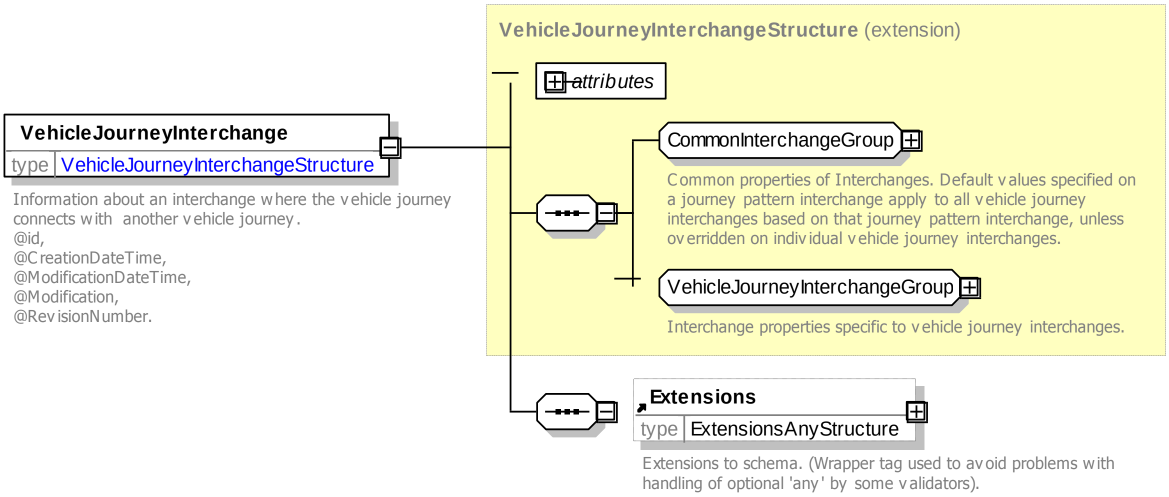


Figure 6-112 – VehicleJourneyInterchange Element

6.8.6.1 VehicleJourney/ VehicleJourneyInterchangeGroup

The ***VehicleJourneyInterchangeGroup*** (*Figure 6-113*) holds elements that are specific to a ***VehicleJourneyInterchange***:

- ***VehicleJourneyRef:*** Optional reference to parent ***VehicleJourney***. Normally not stated as given by containing context, but may be specified when using the Interchange as a stand-alone artefact. If already given by context, this value is ignored**.** (+TXC v2.4).
- ***JourneyPatternInterchangeRef:*** The ***JourneyPatternInterchange*** to which this ***VehicleJourneyInterchange*** corresponds.
- ***InboundVehicleJourneyPatternRef***: The ***VehicleJourney*** of the incoming journey that connects at the interchange.
- ***OutboundVehicleJourneyPatternRef***: The ***VehicleJourney*** of the on-going journey that connects at the interchange.

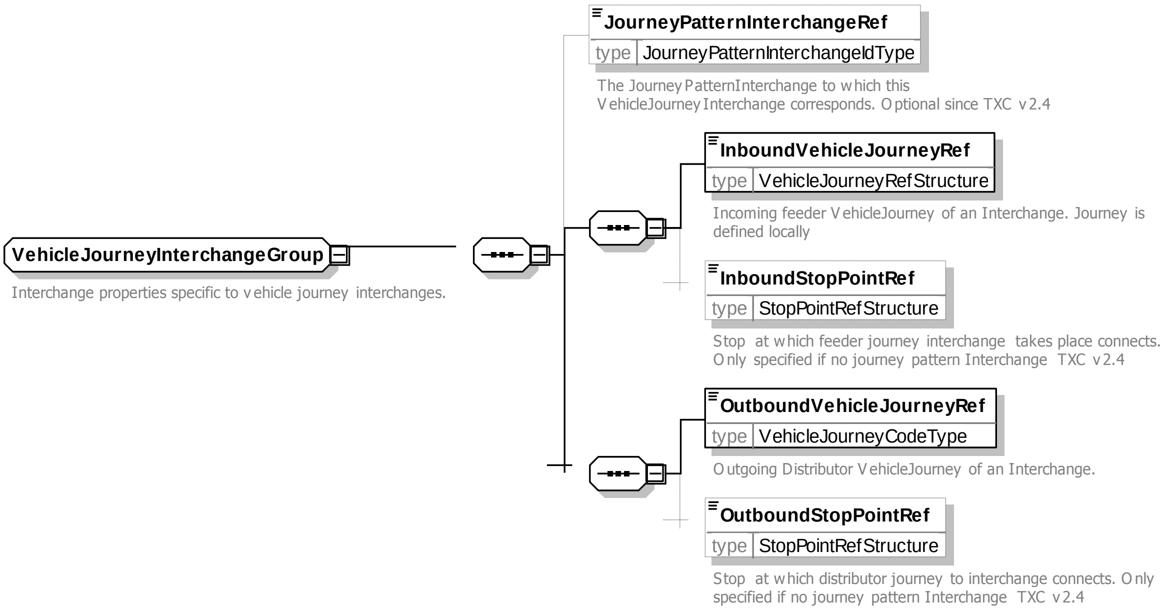


Figure 6-113 – VehicleJourneyInterchangeGroup

6.9 FlexibleService, FlexibleJourneyPattern, FlexibleVehicleJourney

6.9.1 FlexibleService Element

The ***FlexibleService*** element (

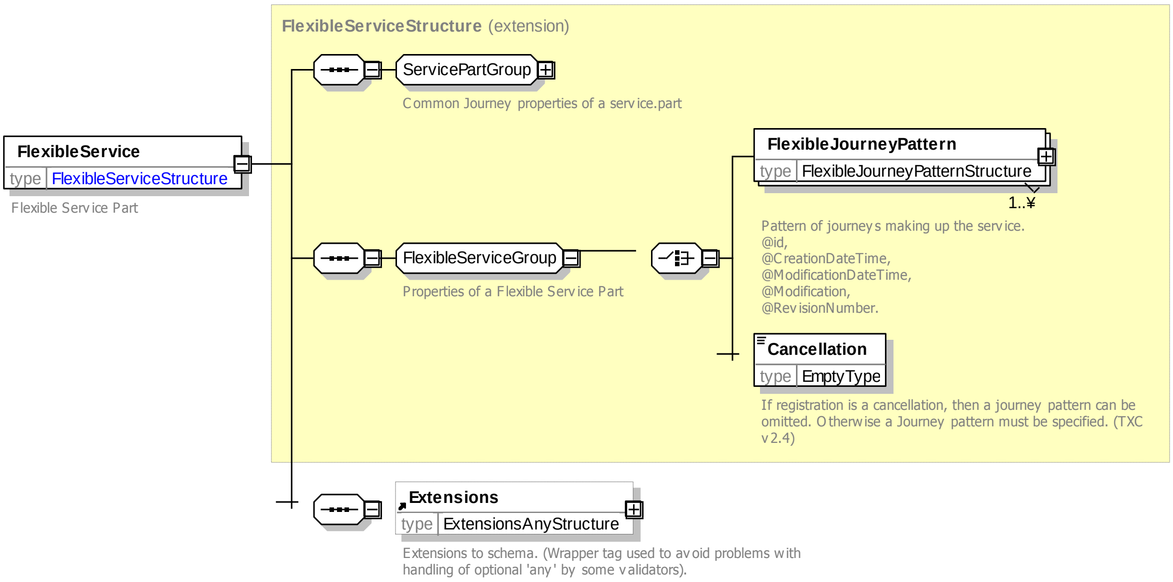


Figure 6-114) describes the flexibly routed component of a ***Service,*** using one or more instances.

- ***ServicePartGroup –*** elements common to standard and flexible journey parts. See earlier.
- ***FlexibleServiceGroup –*** elements specific to ***FlexibleService*** part.
    - ***FlexibleJourneyPattern***: One or more ***FlexibleJourneyPattern*** elements representing the working of the service. See below.
    - ***Cancellation***: If registration is a cancellation, ***FlexibleJourneyPattern*** may be omitted.

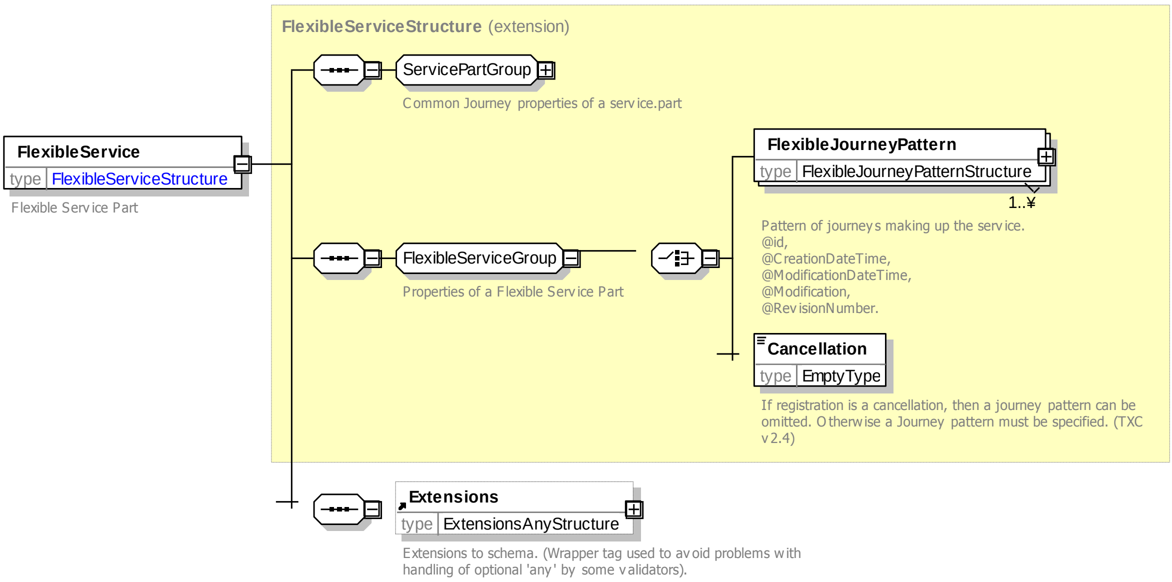


Figure 6-114 – FlexibleService Element

6.9.1.1 FlexibleJourneyPattern Element

The ***FlexibleJourneyPattern*** element (*Figure 6-116*) describes the availability of a flexibly routed journey of a ***Service***. It is made up of two parts:

1. ***CommonJourneyGroup:*** Shared elements common to journey patterns and fixed and flexible vehicle journeys. See ***JourneyPattern / CommonJourneyGroup*** earlier. The ***JourneyPattern*** instances provide default values to use on dependent ***FlexibleVehicleJourney*** instances if no specific override is provided on an individual ***FlexibleVehicleJourney***.
2. ***FlexibleJourneyPatternGroup:*** Elements specific to flexible journey patterns.

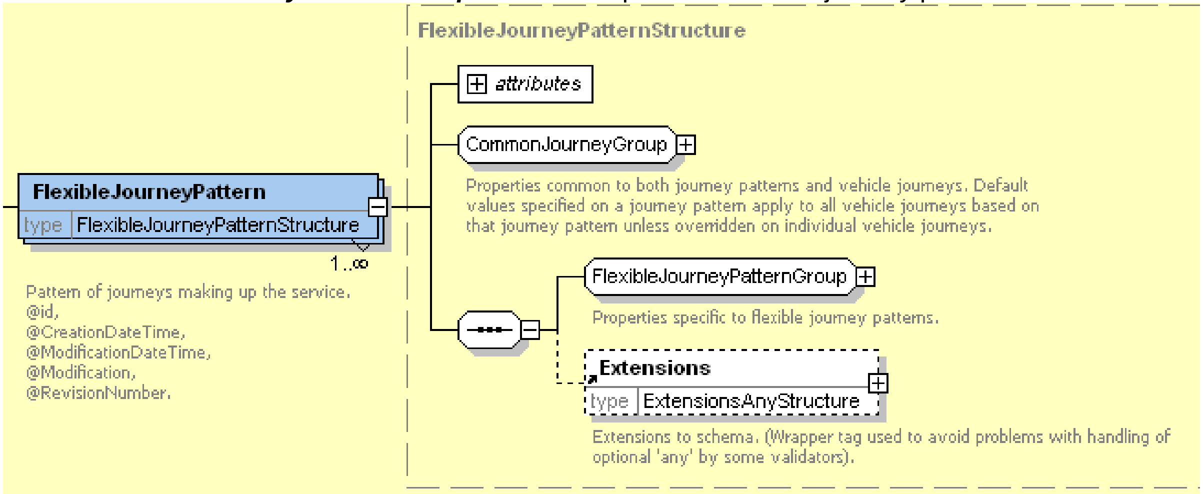


Figure 6-115 – FlexibleJourneyPattern Element

6.9.1.2 FlexibleJourneyPattern / FlexibleJourneyPatternGroup

The ***FlexibleJourneyPatternGroup*** (*Figure 6-116*) holds elements specific to a flexible journey pattern that describes the area of flexible operation and comprises as follows:

- ***FlexibleZones***: Describes the zones that the service covers. See ***FlexibleStopUsage*** below.
- ***FixedStopPoints***: Describes any fixed stops that can be visited by the service. See ***FixedStopUsage*** below.
- ***BookingArangements***: Arrangements for booking the service. See below.

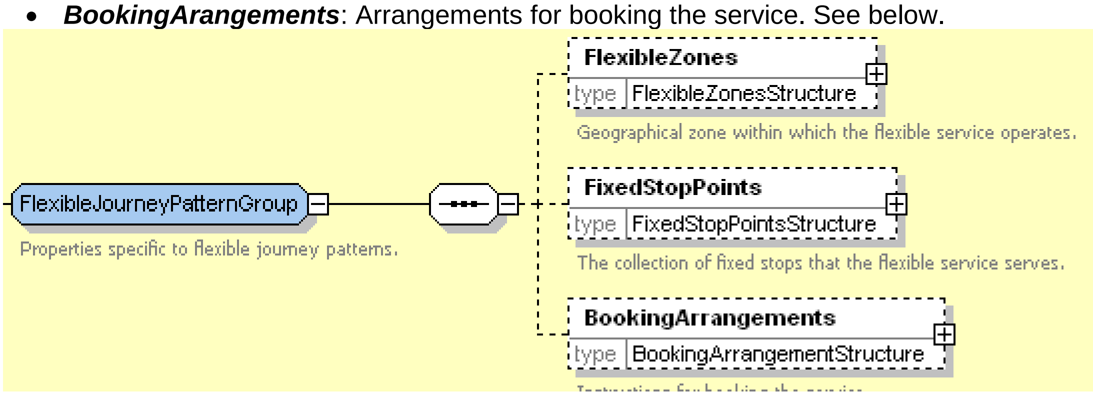


Figure 6-116 – FlexibleJourneyPattern Element

6.9.2 FlexibleService Subelements

6.9.2.1 FlexibleService / FlexibleZones Element

A flexible journey pattern describes the areas and stops covered by a flexible service as two lists: one of flexible zones, and one of fixed stops (*Figure 6-117*).

- ***FlexibleZones***, Comprises a collection of ***FlexibleStopUsage*** instances: each is a ***FlexibleStopUsage*** instance with an activity (e.g. pick up, set down), and a reference to a *NaPTAN* stop of type ***FlexibleZone***.
  - ***Activity***: Activity undertaken by vehicle at stop. See *Table 6-48*. Defaults to pick up and set down.

- *StopPointRef*: *NaPTAN* Stop at which timing link starts or ends.

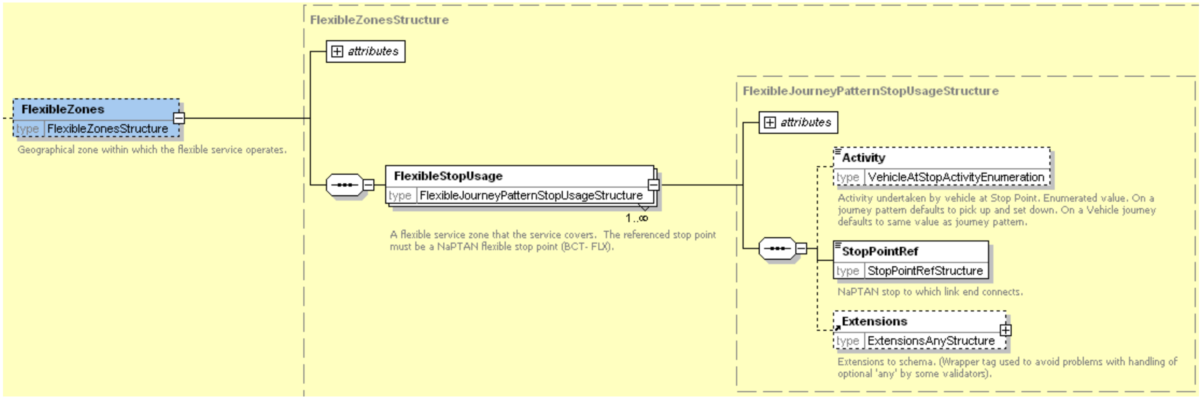


Figure 6-117 – FlexibleZones Element

6.9.2.2 FlexibleService / FixedStopPoints Element

The ***FixedStopPoints***: *Figure 6-118*) is an ordered collection of ***FixedStopUsage*** instances: each is a ***JourneyPatternStopUsageStructure*** (see earlier) instance with an activity (e.g. pick up, set down), and a reference to a *NaPTAN* fixed stop, i.e. of any type ***such as MarkedPoint,*** other than ***FlexibleZone.***

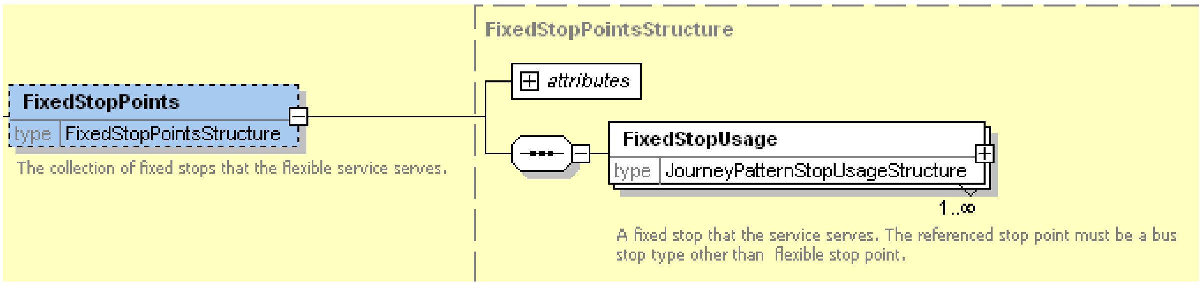


Figure 6-118 – FlexibleService / FixedStopPoints Element

6.9.2.3 FlexibleVehicleJourneyGroup / BookingArrangements Element

The ***BookingArrangements*** element (*Figure 6-119*) describes the booking arrangements for the flexible service:

- ***Description***: Text description of booking process.
- ***Phone***: Phone number by which to make bookings. See ***TelephoneContactStructure.***
- ***Email:*** Email address to which to make bookings.
- ***Address***: Postal address by which to make bookings. See ***PostalAddressStructure.***
- ***WebAddress:*** URL of online web site by which make bookings.
- ***AllBookingsTaken:*** Whether all bookings are taken. Default is true.

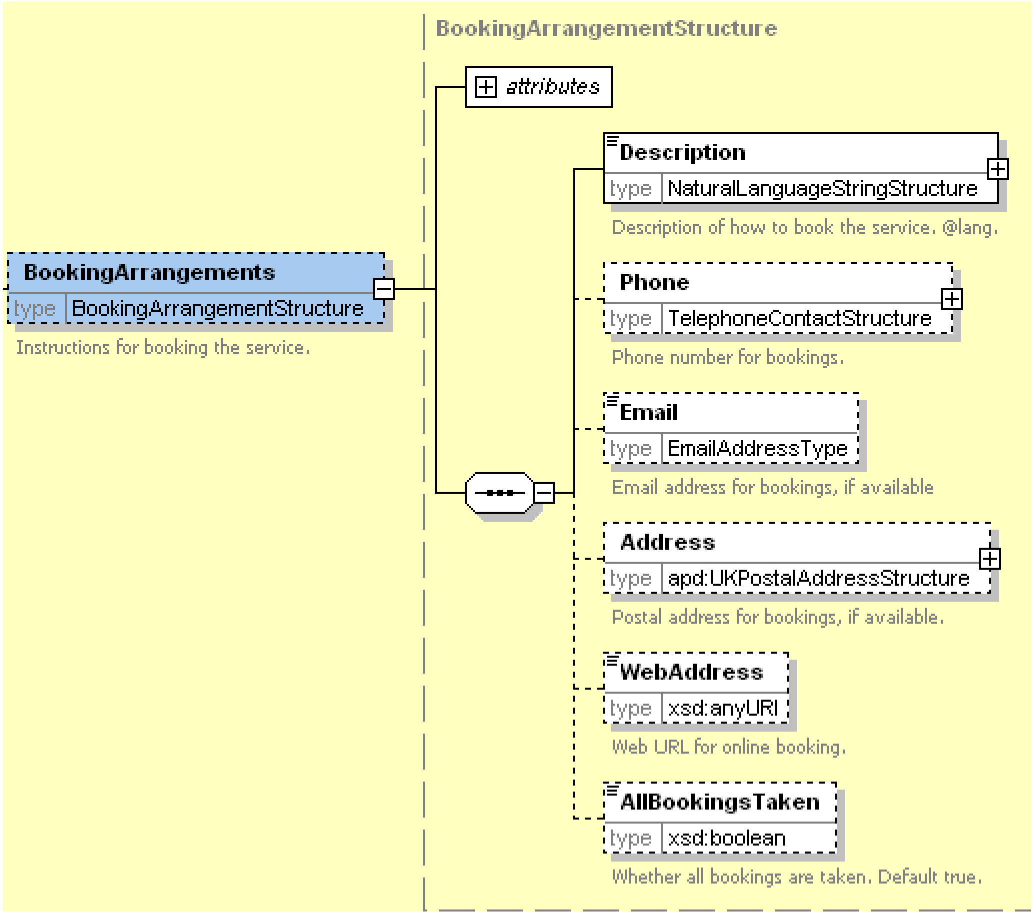


Figure 6-119 – FlexibleVehicleJourney / BookingArrangements Element

6.9.3 FlexibleVehicleJourney Element

The ***FlexibleVehicleJourney*** element (*Figure 6-120*) describes the availability of a flexible journey. It adds time information to a ***FlexibleJourneyPattern*** instance. A ***FlexibleVehicleJourney*** comprises a number of elements; the elements fall into three groups:

1. ***CommonJourneyGroup:*** Shared elements common to journey patterns and vehicle journeys (See ***JourneyPattern / CommonJourneyGroup*** earlier).
2. ***VehicleJourneyGroup:*** Elements specific to both fixed and flexible vehicle journeys (See ***VehicleJourney / VehicleJourneyGroup*** earlier).
3. ***FlexibleVehicleJourneyGroup:*** Elements specific to flexible route vehicle journeys: See ***FlexibleVehicleJourneyGroup*** / ***FlexibleServiceTimes*** below.

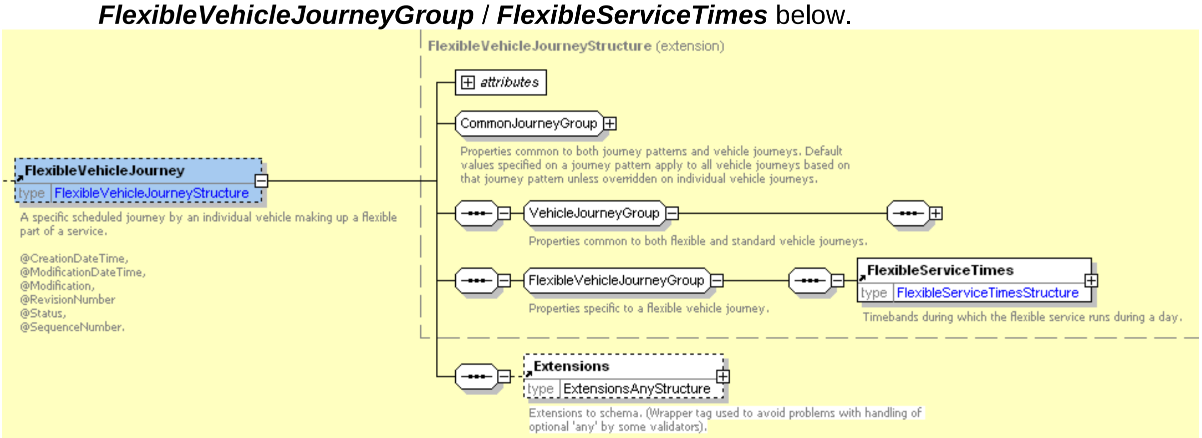


Figure 6-120 – FlexibleVehicleJourney

6.9.3.1 FlexibleVehicleJourneyGroup / FlexibleServiceTimes Element

The ***FlexibleServiceTimes*** element (*Figure 6-121*) describes the operational days of the service. ***FlexibleServiceTimes*** may either be:

- ***AllDayService***: Indicating the service runs all day, or
- ***PeriodsOfOperation***: A collection of at least one ***ServicePeriod*** element, made up of:
  - ***StartTime***: Time at which time band starts.
  - ***EndTime***: Time at which time band ends.

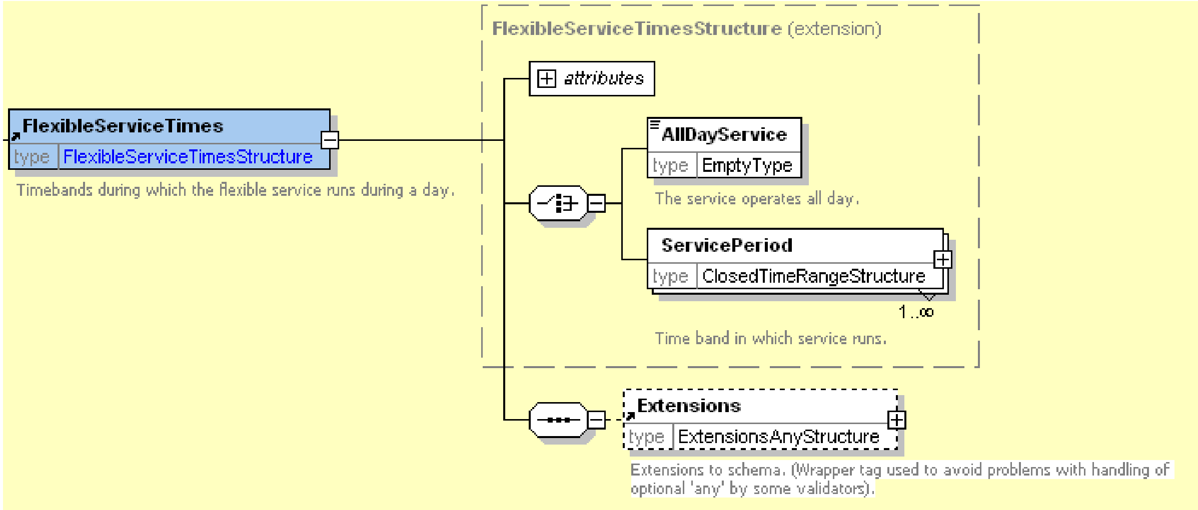


Figure 6-121 – FlexibleVehicleJourney / FlexibleServiceTimes Element

6.10 ConnectingVehicleJourney Element

A ***ConnectingVehicleJourney*** (*Figure 6-122*) describes a connecting journey. It allows a subset of a vehicle journey's properties to be specified. A ***ConnectingVehicleJourney*** comprises a number of elements; the elements fall into two groups:

4. ***CommonJourneyGroup:*** Shared elements common to journey patterns and vehicle journeys. See ***JourneyPattern*** / ***CommonJourneyGroup*** earlier. Allows individual properties to be overridden on a vehicle journey: if not specified the property from the journey pattern will be used.
5. ***ConnectingVehicleJourney:*** Elements specific to connecting vehicle journeys, both fixed and flexible.

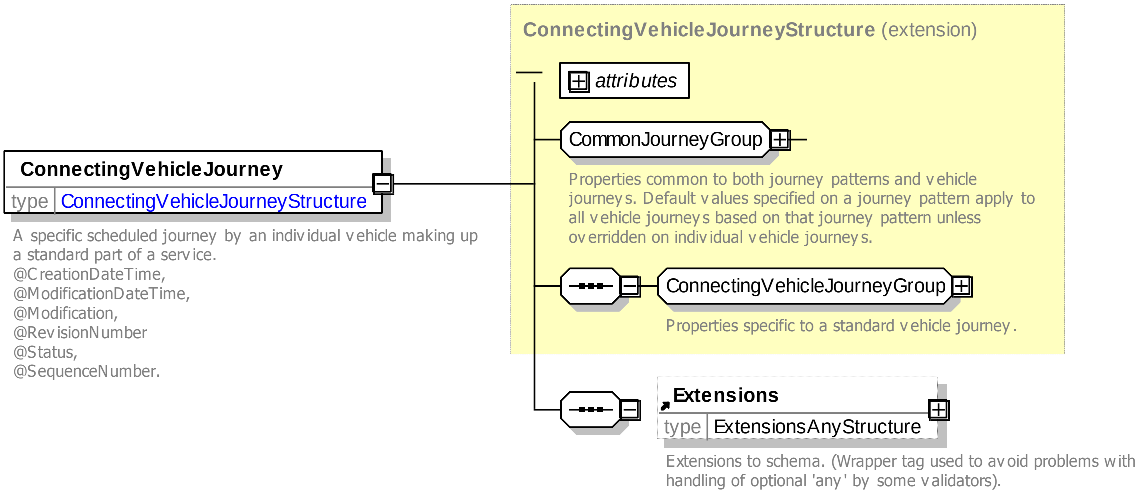


Figure 6-122 – ConnectingVehicleJourney Element

6.10.1 VehicleJourney / ConnectingVehicleJourney Group

The ***ConnectingVehicleJourneyGroup*** (*Figure 6-103*) holds elements that are specific to ***ConnectingVehicleJourneyGroup*** instances:

- ***VehicleJourneyCode***: A unique code that can be used to identify the ***VehicleJourney***.
- ***ServiceRef***: The ***Service*** to which the ***VehicleJourney*** belongs.
- ***RegistrationRef***: Reference to a registration of a journey
- ***AnnotatedOperatorRef:*** Reference to an operator. of journey, includes name
- ***AnnotatedLineRef***: Reference to the ***Line*** that the ***VehicleJourney*** serves. Includes ***LineName*** the Public identifier of the ***Line.***
- ***ConnectingTimesGroup:*** Describes the times for a ***ConnectingVehicleJourney***. See below.
- ***VehicleJourneyInterchange:*** Interchanges where the vehicle journey connects with another vehicle journey.
- ***Note:*** Any additional notes on the ***VehicleJourney***.

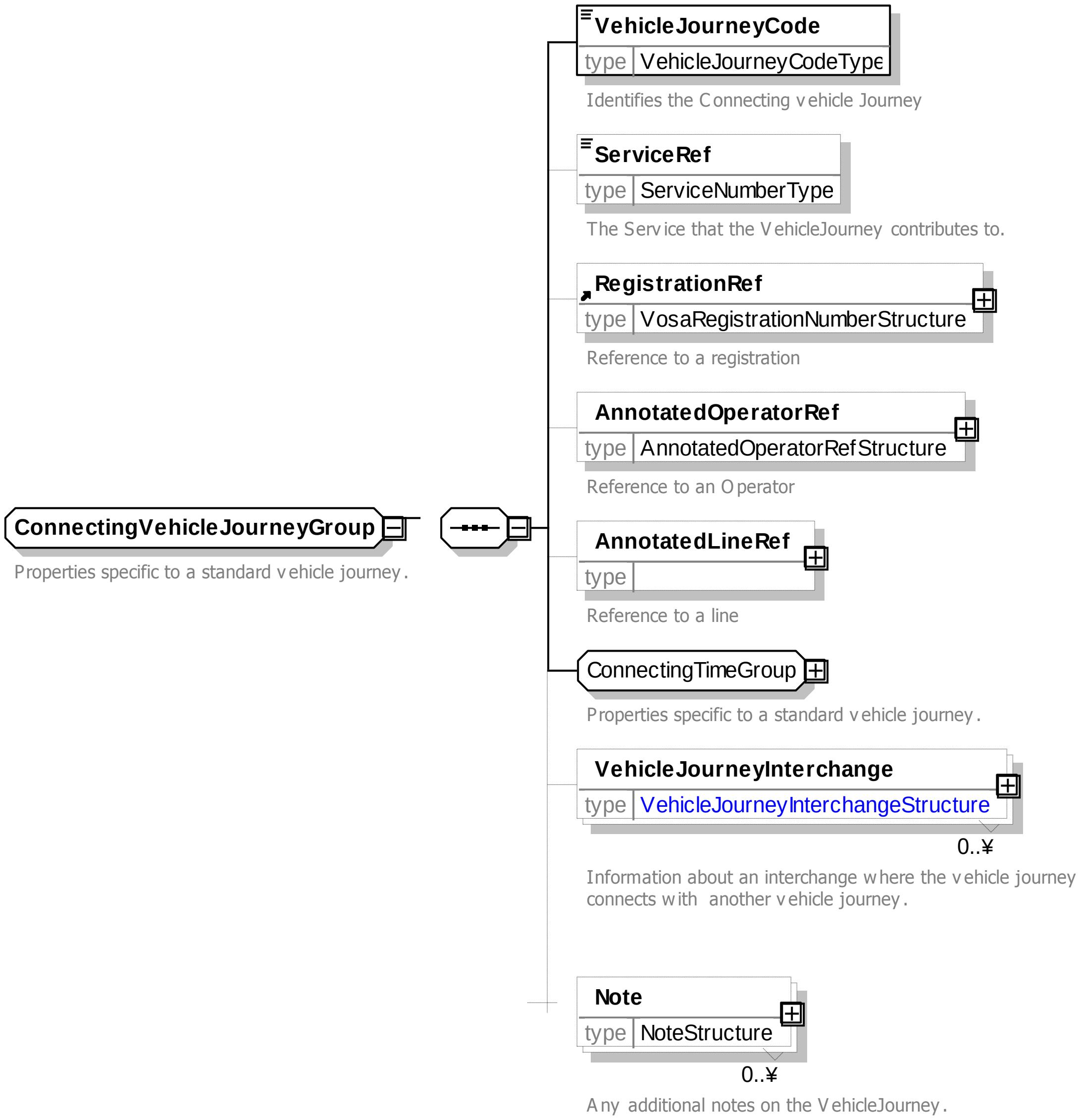


Figure 6-123 – ConnectingVehicleJourney Element

In this section we describe the schema elements used to specify operational dates and times in *TransXChange*. These are common to both Flexible and Standard services. See also the earlier section 3.17.8 on *Modelling Operational days*, which sets out the rules used for combining the various day type and date elements.

6.10.1.1 ConnectingVehicleJourney / ConnectingTimesGroup

The ***ConnectingTimesGroup*** (*Figure 6-124*) describes the times for a ***ConnectingVehicleJourney***.

- ***ArrivalTime:*** Time of arrival at destination stop of the ***VehicleJourney***. Used for feeder journeys.
- ***ArrivalDayShift:*** Whether the ***ArrivalTime*** is the same day (0 or omit,) the next day (+1) or the Previous day. (-1) This allows Journeys that start or end after midnight to be included in a particular day type as expressed by the profile.
- ***DepartureTime:*** Time of departure from origin stop of the ***VehicleJourney***.
- ***DepartureDayShift:*** Whether the ***DepartureTime*** is the same day (0 or omit,) the next day (+1) or the Previous day. (-1) This allows Journeys that start or end after midnight to be included in a particular day type as expressed by the profile.
- ***Frequency***: Describes service frequency for frequency based services. See below.

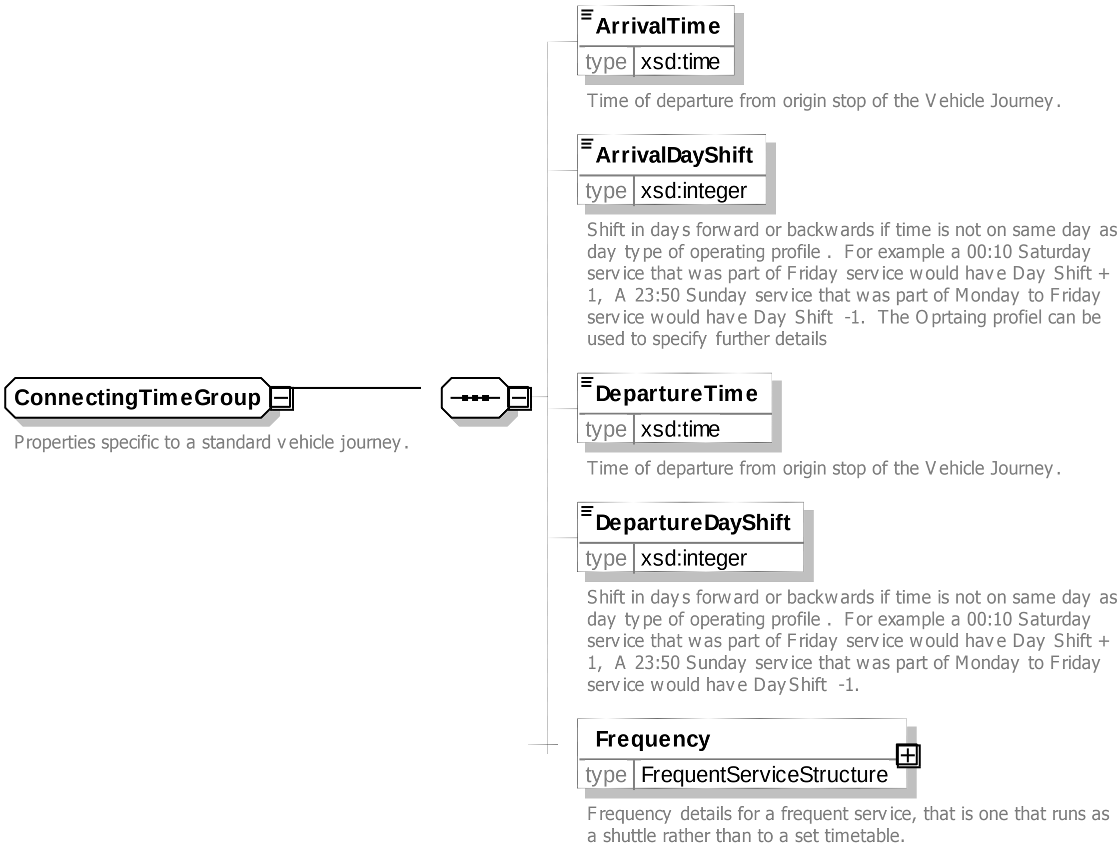


Figure 6-124 – ConnectingVehicleJourney / ConnectingTimesGroup

6.11 OperatingProfile Element

The ***OperatingProfile*** element (*Figure 6-125*) specifies on which days a service operates. An ***OperatingProfile*** can be specified on both a ***VehicleJourney,*** a ***JourneyPattern*** and on a ***Service;*** the ***VehicleJourney*** values override those of the ***JourneyPattern*** or ***Service***. It is made up of two groups:

1. *Normal* operating profile group: describes normal regular behaviour.
2. *Special* operating profile group: describes behaviour on bank holidays and other exceptional days.

If no ***OperatingProfile*** profile is specified a default value will be used. *This assumes the service runs Monday to Friday every day of the year.* This default will be shown by the Publisher on the ***Registration*** particulars and Matrix footnote. (TXC v2.4)

6.11.1 Normal OperatingProfileGroup

The ***OperatingProfile*** *normal* elements describe the regular operation of the service and comprise the following elements:

- ***RegularDayType:*** specifies the days on which the service normally runs. See below. Defaults to ***MondayToSunday***.
- ***PeriodicDayType:*** qualifies the ***RegularDayType*** days with any specific weeks of the month that the service runs. It is logically 'ANDed' with ***RegularDayType***, so that you may specify for example '*Wednesdays, first and third weeks of the month'*.
- ***ServicedOrganisationDayType:*** Specifies that the service runs or does not run on the working days or holidays of a nominated organisation such as a school or Local Education Authority. See ***ServicedOrganisation*** days below. ***ServicedOrganisationDayType*** is 'ANDed' with ***RegularDayType*** and any ***PeriodicDayType*** values.

6.11.2 Special OperatingProfileGroup

The ***OperatingProfile*** *special* elements describe exceptions to the normal days of operation and comprise the following elements:

- ***SpecialDaysOperation***: Describes the specific dates (other than standard bank holiday types) when the service will operate differently from its normal service. ***DaysOfOperation*** and ***DaysOfNonOperation*** can be specified separately. See below.
- ***BankHolidayOperation***: Describes how the service will operate on bank holidays. ***DaysOfOperation*** and ***DaysOfNonOperation*** can be specified separately. See below.

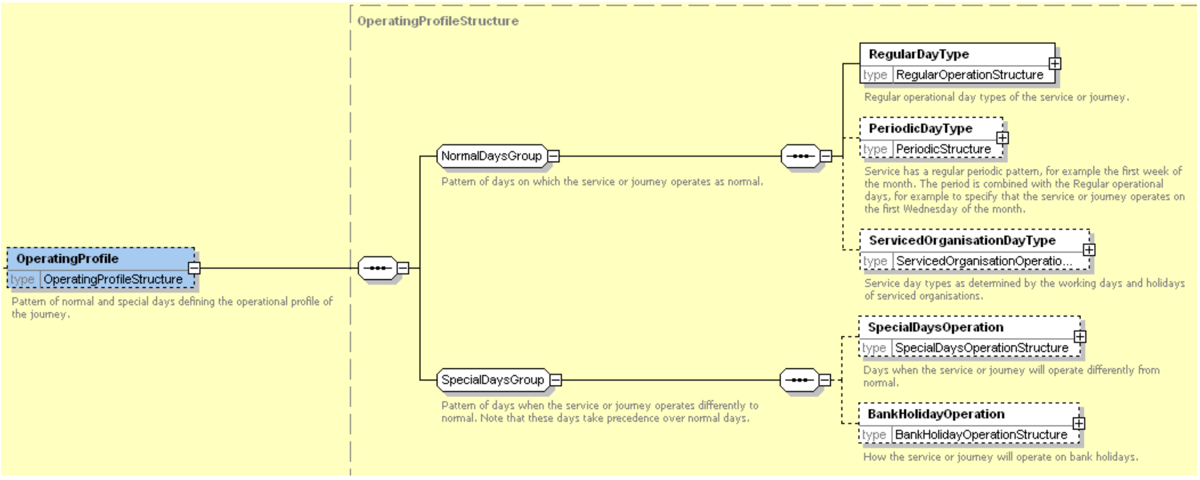


Figure 6-125 – OperatingProfile Element

6.11.3 OperatingProfile Subelements

6.11.3.1 OperatingProfile / RegularDayType Element

The ***RegularDayType*** element (*Figure 6-126*) specifies the normal days of operation of the associated service, journey pattern or vehicle journey. It comprises either:

- ***DaysOfWeek***: Week days on which service operates. See below.
- ***HolidaysOnly***: Service only runs on holidays specified by ***OperatingProfile*** *special* elements.

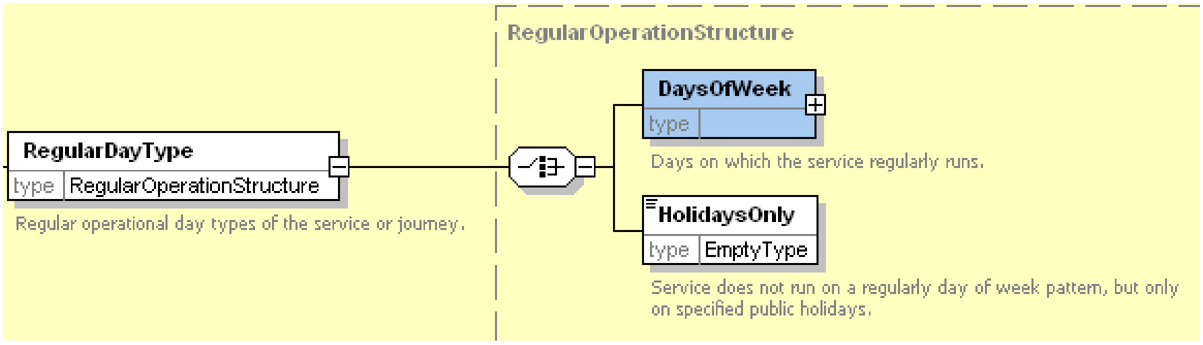


Figure 6-126 – OperatingProfile / RegularDayType Element

6.11.3.2 OperatingProfile / RegularDayType / DaysOfWeek Element

The ***DaysOfWeek*** element specifies any combination of day types using a ***DayGroup*** structure (*Figure 6-127*). It allows any meaningful combination of the following:

- Week days:
  - ***Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday***
- Groups of days:
  - ***MondayToFriday, MondayToSaturday, MondayToSunday, NotSaturday***
  - ***Weekend:*** Saturday and Sunday.

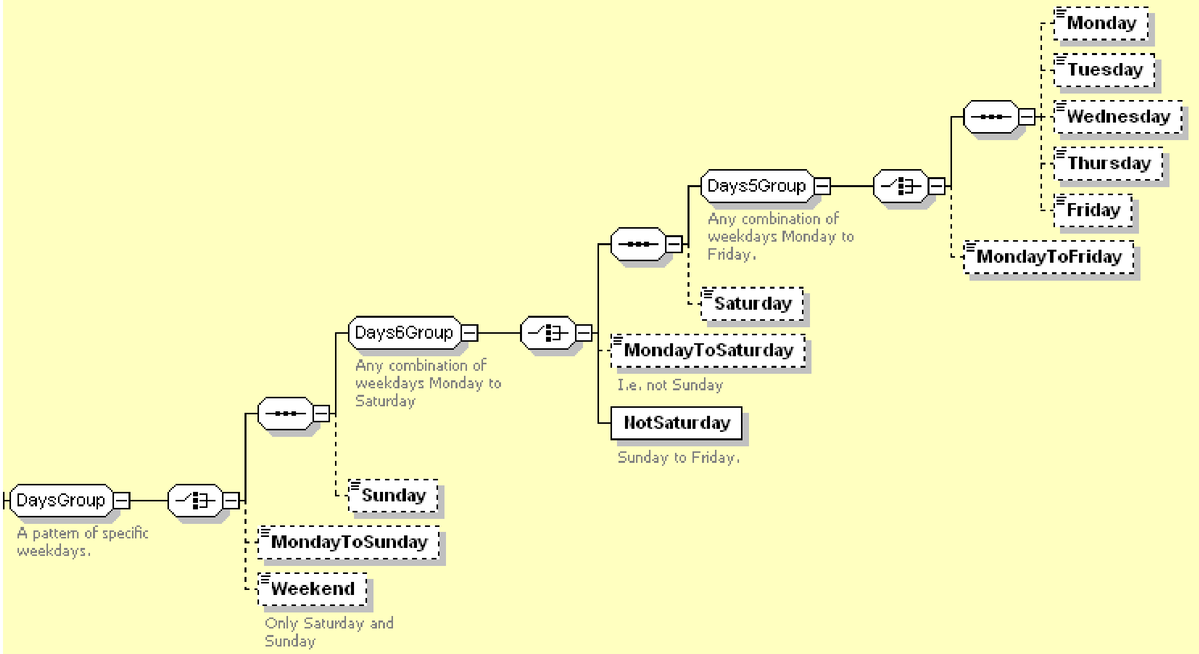


Figure 6-127 – OperatingProfile / DaysOfWeek Element

6.11.3.3 OperatingProfile / PeriodicDayType / WeekOfMonth Element

The ***PeriodicDayType / WeekOfMonth*** element (*Figure 6-128*) specifies any combination of week types within a month, using up to four ***WeekNumber*** elements, i.e. any subset of four elements out of the set of numbers 1, 2,3,4,5. The week numbers are combined with the day type, for example: *'First Wednesday in the month'*.

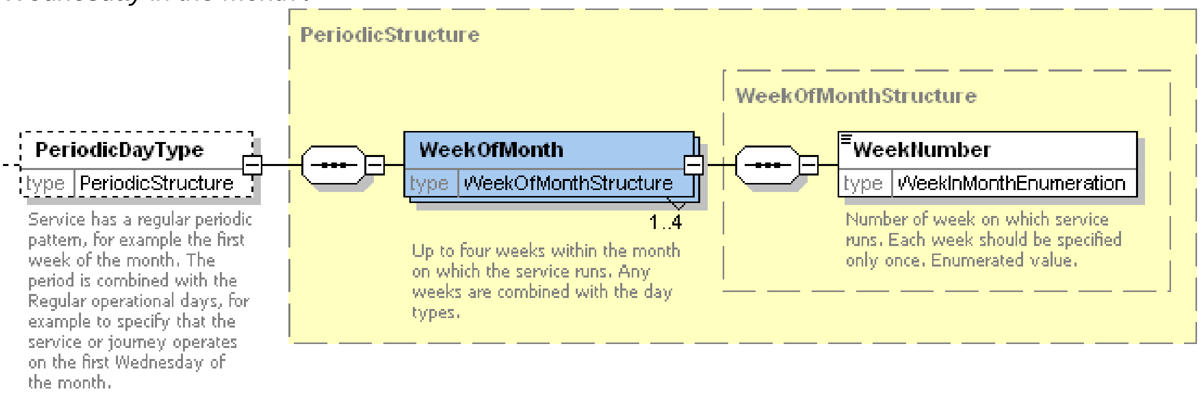


Figure 6-128 – OperatingProfile / WeekOfMonth Element

6.11.3.4 SpecialDaysOperation Element: DaysOfOperation, DaysOfNonOperation

The ***SpecialDaysOperation*** element (*Figure 6-129*) describes specific dates when a service does or does not operate (other than Bank Holiday day types), and comprises two collections of ***DateRange*** elements, wrapped in ***DaysOfOperation*** and ***DaysOfNonOperation*** elements respectively. If conflicting dates are specified, days of non-operation are given precedence.

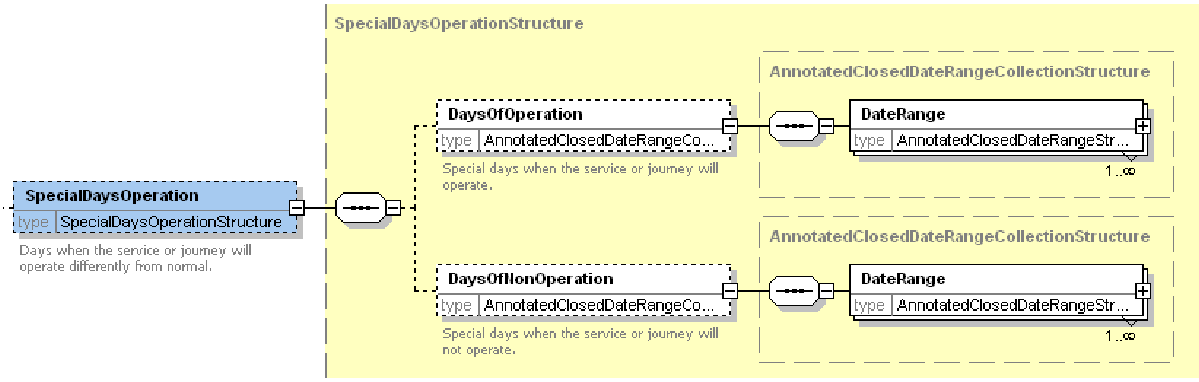


Figure 6-129 – OperatingProfile / SpecialDaysOfOperation Element

6.11.3.5 DateRange

The ***DateRange*** element (*Figure 6-130*) describes a period. Each range is specified with the following:

- ***StartDate:*** Inclusive date on which period starts.
- ***EndDate***: Inclusive date on which period ends.
- ***Note***: Annotation about period.

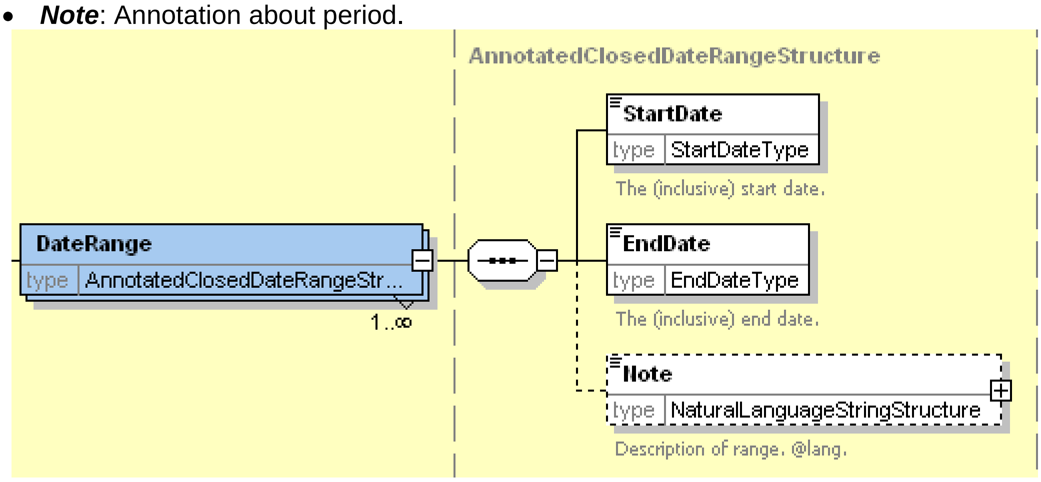


Figure 6-130 – DateRange Element

6.11.3.6 OperatingProfile / BankHolidayOperation

The ***BankHolidayOperation*** element (*Figure 6-131*) describes how the service does or does not operate on bank holidays, and comprises two collections of ***BankHolidayStructure*** elements, wrapped in ***DaysOfOperation*** and ***DaysOfNonOperation*** elements respectively. If conflicting dates are specified, days of non-operation are given precedence.

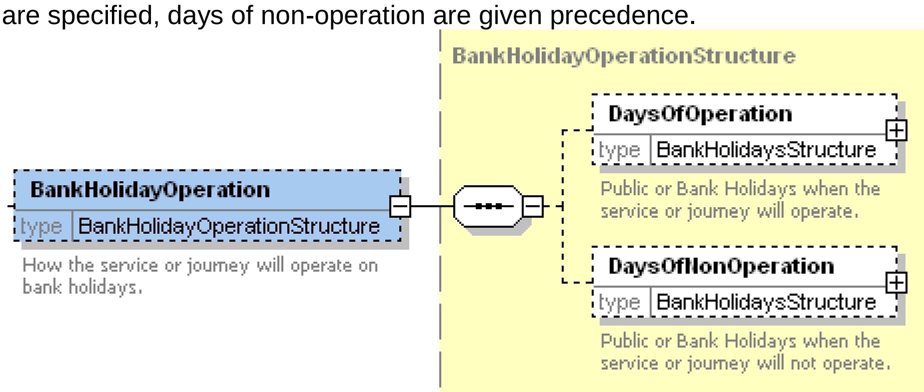


Figure 6-131 – OperatingProfile / BankHolidayOperation Element

6.11.3.7 OperatingProfile / BankHoliday Elements

Holiday day types are explicitly enumerated using the ***BankHolidayOperationStructure*** (see Figure 6-132), which allows individual holidays or combinations of holidays to be enumerated. The ***AllBankHolidaysGroup*** is used to denote all Bank Holidays in the country in which the service runs.

- ***EarlyRunOffGroup*** can be used to indicate special services for Christmas and New Year's Eve.
  - ***EarlyRunOffDays*** : any Early Run off Day
  - ***ChristmasEve***, ***NewYearsEve***: Specific early run off days.

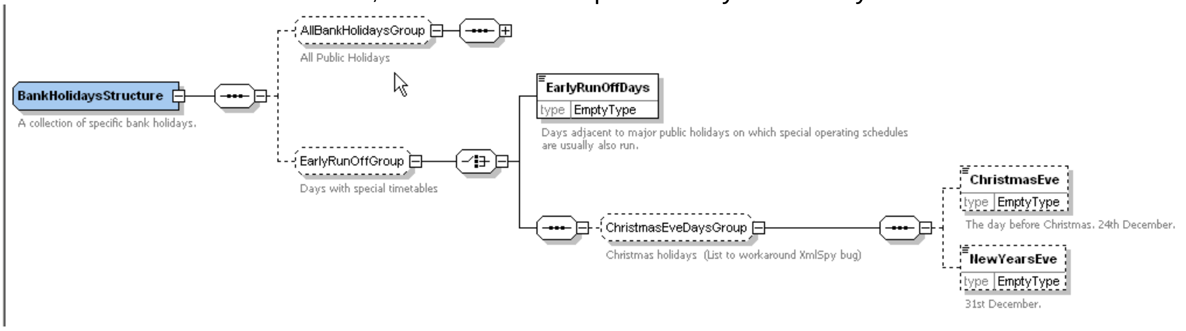


Figure 6-132 – OperatingProfile / Bank Holidays Element

6.11.3.8 OperatingProfile / AllBankHolidayGroup

The ***AllBankHolidaysGroup*** specifies standard and custom bank holidays

- A special element ***AllBankHolidays*** is used to denote all standard Bank Holidays in the country in which the service runs. See *Table 6-54.*
  - The ***HolidayMondays*** element can be used to denote all the summer Bank holiday Mondays.
- ***OtherBankHoliday*** can be used to specify additional special holidays.
  - ***Description***: specifies the name of the element can be used to denote all the summer Bank holiday Mondays.
  - ***Date:*** specifies the date on which the holiday occurs

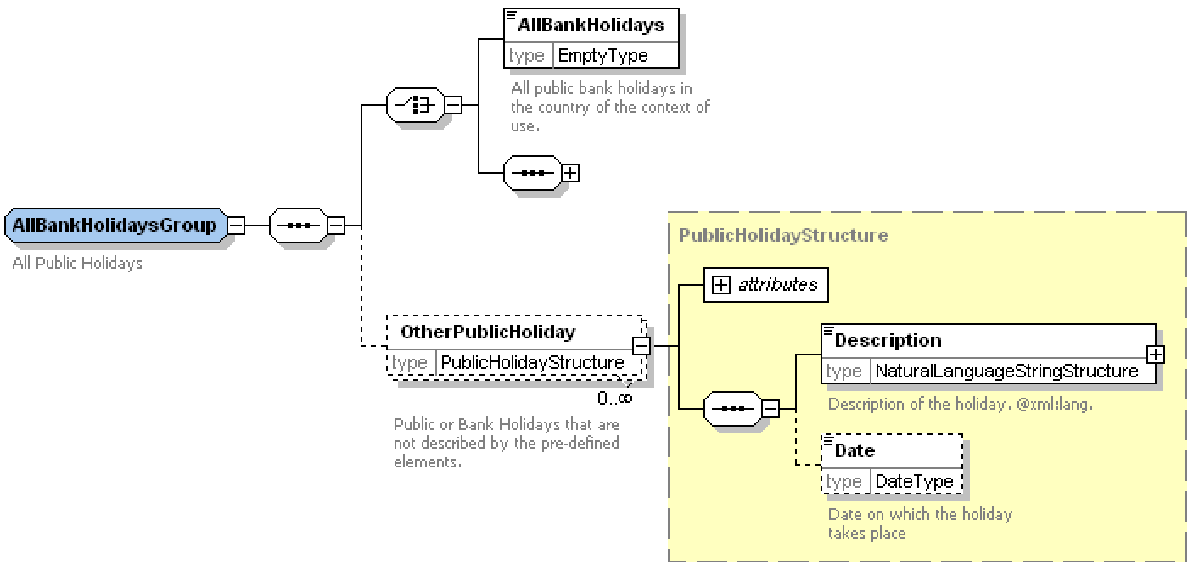


Figure 6-133 – OperatingProfile / AllBankHolidaysGroup

6.11.3.9 OperatingProfile / AllBankHoliday Element

A special element ***AllBankHolidays*** is used to denote all standard Bank Holidays in the country in which the service runs. See *Figure 6-133* and *Table 6-54.*

- The ***HolidayMondays*** element can be used to denote all the summer Bank holiday Mondays.
- ***Christmas*** can be used to indicate special services for actual ***ChristmasDay*** (strictly the 25th December) and ***BoxingDay*** (strictly the 26th December).
- The ***HolidayMondays*** element can be used to denote all the summer Bank holiday Mondays.
- The ***AllHolidaysExceptChristmas*** element can be used to denote all the Bank holidays in the year except for ***ChristmasDay*** and ***BoxingDay***.
- ***DisplacementHolidays*** can be used to indicate special services for Public holidays that are awarded when calendar based holidays such as Christmas Day, Boxing Day or New Year's Eve fall at a weekend so a compensating weekday, usually a Monday or Friday, is also made a public holiday. Sometimes different timetables are used for the Displacement Holiday from those that would be used for the actual day itself.

Group		Subgroup	England & Wales	Scotland	MM/DD	
AllBankHolidays	***AllHolidays Except Christmas***	***Holidays***	***NewYearsDay***	***NewYearsDay***	***01/01***	***2.0***
				Jan2ndScotland	***01/02***	***2.0***
			GoodFriday	***GoodFriday***	***var***	***2.0***
				StAndrewsDay	***11/30***	***2.4***
		Holiday Mondays	***EasterMonday***	***EasterMonday***	***var***	***2.0***
			MayDay	***MayDay***	***var***	***2.0***
			SpringBank	***SpringBank***	***var***	***2.0***
			LateSummerHoliday NotScotland	***AugustBankHoliday Scotland***	***var***	***2.0***
		Christmas	***ChristmasDay***	***ChristmasDay***	***12/25***	***2.0***
			BoxingDay	***BoxingDay***	***12/26***	***2.0***
		Displacement Holidays	***ChristmasDayHoliday***	***ChristmasDayHoliday***	***var***	***2.1***
			BoxingDayHoliday	***BoxingDayHoliday***	***var***	***2.1***
			NewYearsDayHoliday	***NewYearsDayHoliday***	***var***	***2.1***
				Jan2ndScotlandHoliday	***var***	***2.4***
				StAndrewsDayHoliday	***var***	***2.4***
EarlyRunOff		***-***	***ChristmasEve***	***ChristmasEve***	***12/23***	***2.0***
		-	***NewYearsEve***	***NewYearsEve***	***12/31***	***2.0***

Table 6-54 – AllBankHolidays by Country

*** Not official but often observed by Scottish banks and Retailers**

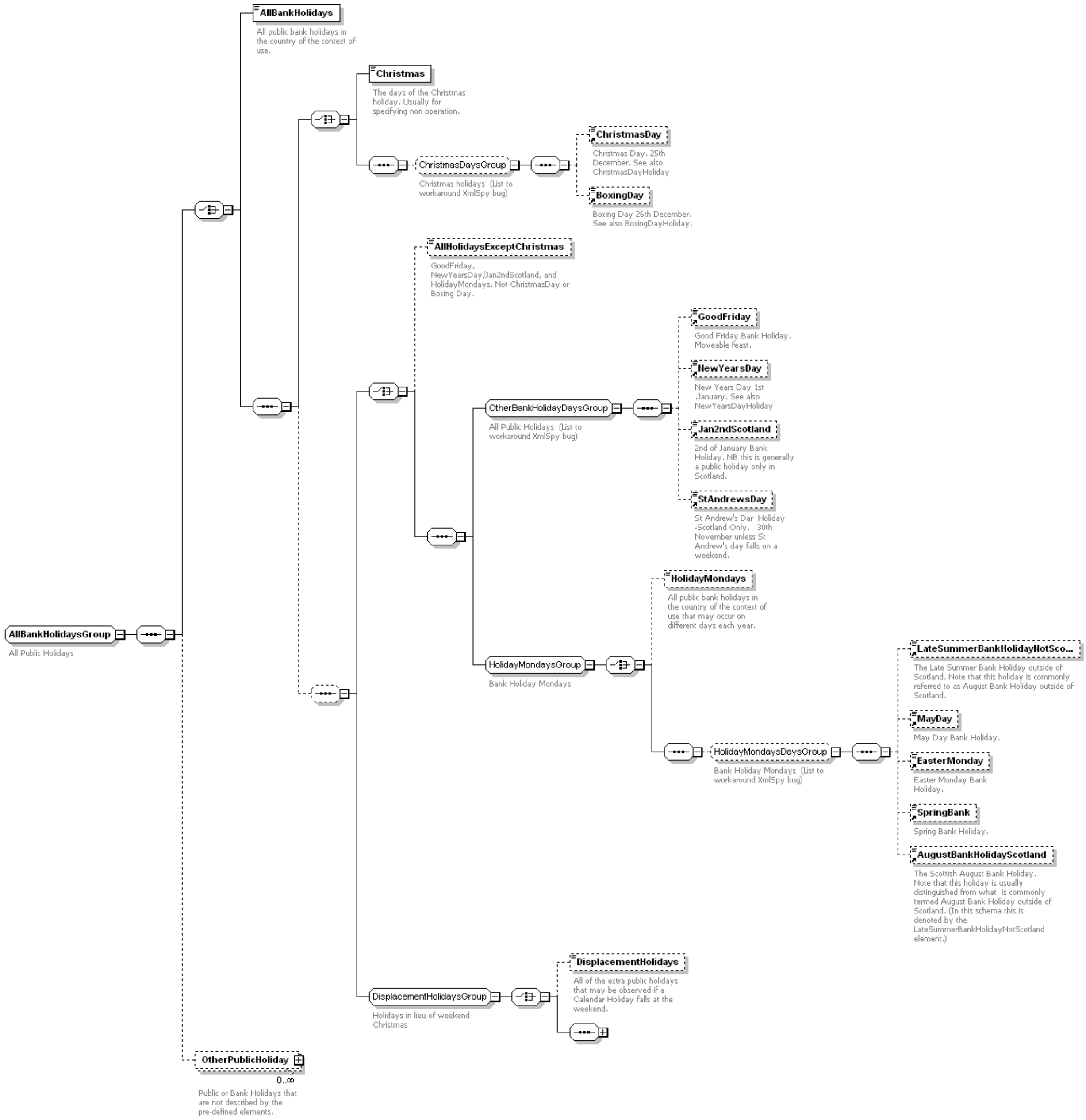


Figure 6-134 – OperatingProfile / All Bank Holidays Group

6.12 ServicedOrganisation Element

Operational days can also be specified in terms of the working days or holidays of specified organisations, for example schools. The ***ServicedOrganisation*** element is used to define the organisations covered, and to specify their working and non-working days.
A *TransXChange* document may contain a collection of ***ServicedOrganisation*** definitions.
Each ***ServicedOrganisation*** definition (*Figure 6-135*) comprises:

- ***OrganisationCode***: Identifier of the ***ServicedOrganisation***.
- ***PrivateCode***: Alternative code to support interoperability (+TXC v2.4)
- ***Name***: Name of the ***ServicedOrganisation***.
- ***OrganisationContactGroup***: Contact details for organisation. See below. (+TXC v2.4)
- ***ServicedOrganisationClassification***: Indication of the nature of an organisation. See *Table 6-55* below. (+TXC v2.4)

Value	Description
school	School for which services will vary if open
office	Office for which services will vary if open
retailSite	Mall or major shop for which services will vary if open
touristAttraction	POI or tourist attraction for which services will vary if open
market	Street market for which services will vary if open
factory	Factory or Works for which services will vary if open
college	College or University for which services will vary if open
military	Military base for which services will vary if open

Table 6-55 – Allowed values for Serviced Organisation Type

- ***ServicedOrganisationDaysGroup***: Calendar for the Organisation
  - ***WorkingDays***: The working days of the ***ServicedOrganisation***, for example a LEA's terms.
  - ***Holidays***: The non-working days of the ***ServicedOrganisation***, for example a LEA's holidays.
- ***ParentServicedOrganisationRef***: Identifier of another ***ServicedOrganisation*** that is the element's parent. References should be acyclic. Working days and holidays specified for a parent are used as defaults for all child organisations, unless specifically overridden on the child instance.
- ***LocalityGroup***: a calendar for the Organisation
  - ***NptgLocalityRef***: The NPTG Locality with which the ***ServicedOrganisation***, is associated

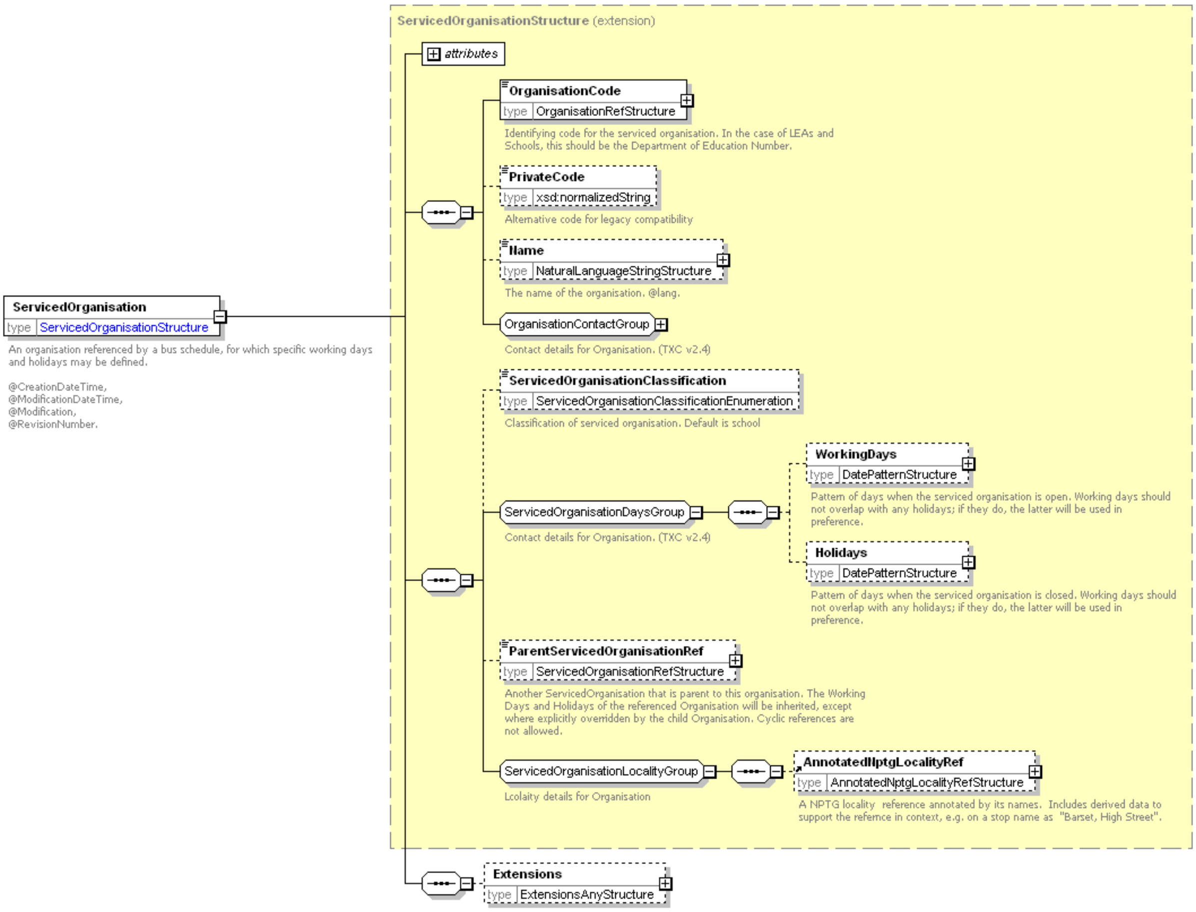


Figure 6-135 – ServicedOrganisation Element

6.12.1 ServicedOrganisation Subelements

6.12.1.1 ServicedOrganisation / DatePattern Element

The ***DatePattern*** element (*Figure 6-136*) specifies a group of one or more non-contiguous periods as a collection of date ranges. See Modelling operation days for precedence of overlapping dates.

- ***DateRange***: A collection of one or more open-ended date ranges, and any number of date exceptions.
  - ***StartDate***: The (inclusive) start date. If omitted, the range start is open-ended, that is, it should be interpreted as "since the beginning of time".
  - ***EndDate***: The (inclusive) end date. If omitted, the range end is open-ended, that is, it should be interpreted as "forever". If it is the same as the Start date it specifies a single day.
- ***Description***: A description or name for the period. E.g. "*Easter Ter*m". (+TXC v2.4)
- ***Provisional***: Whether the date is provisional or firm. (+TXC v2.4)
- ***DateExclusion***: Individual dates within the period which should be omitted.(- TXC v2.4 THIS ELEMENT SHOULD NO LONGER BE USED: Simply use another Date Range instead).
  -
- ***DateClassification***: A classification of the DateRange. (+TXC v2.4). See Table 6-56 .

Value	**Description**
term	Term time of an educational establishment.
working	Working day of an organisation.
inset	Inset day of an educational establishment.
holiday	Non-working day of any organisation.
other	Other day.

Table 6-56 – Allowed values for DateClassification

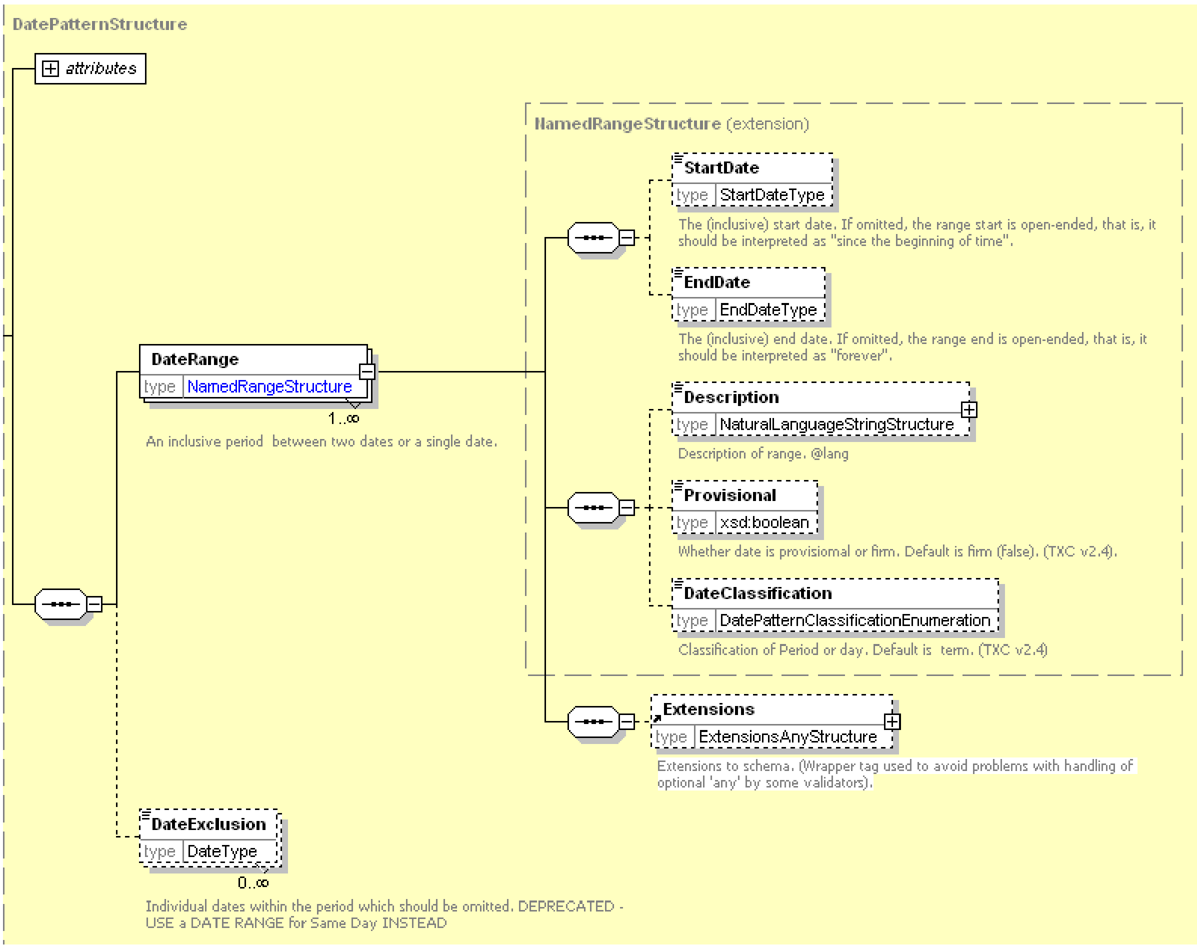


Figure 6-136 – ServicedOrganisation / Date Pattern

6.13 Calendar Element (TXC v2.4)

The ***Calendar*** element provides a list of dates on which specific holiday day types occur. This can be used to resolve an undated vehicle journey to an actual date (+TXC v2.4).
A *TransXChange* document may contain a collection of ***ServicedOrganisation*** definitions.
Each ***Calendar*** (Figure 6-137) comprises:

- ***id***: Identifier of the ***Calendar***.
- ***CalendarPeriod***: Period within which the ***Calendar*** applies.
  - ***StartDate:*** Start date of ***Calendar.***
  - ***EndDate:*** End date of ***Calendar.***
- ***Name:*** *Name of **Calendar.***
- ***OperatingDays:*** One or more ***OperatingDays*** contained in the ***Calendar***.
  - ***OperatingDay: OperatingDay*** contained in ***Calendar.*** See below**.**
- ***DayTypes:*** One or more ***DayType a*** contained in the ***Calendar:***
  - ***DayType:*** Day type defined by ***Calendar.*** See below**.**

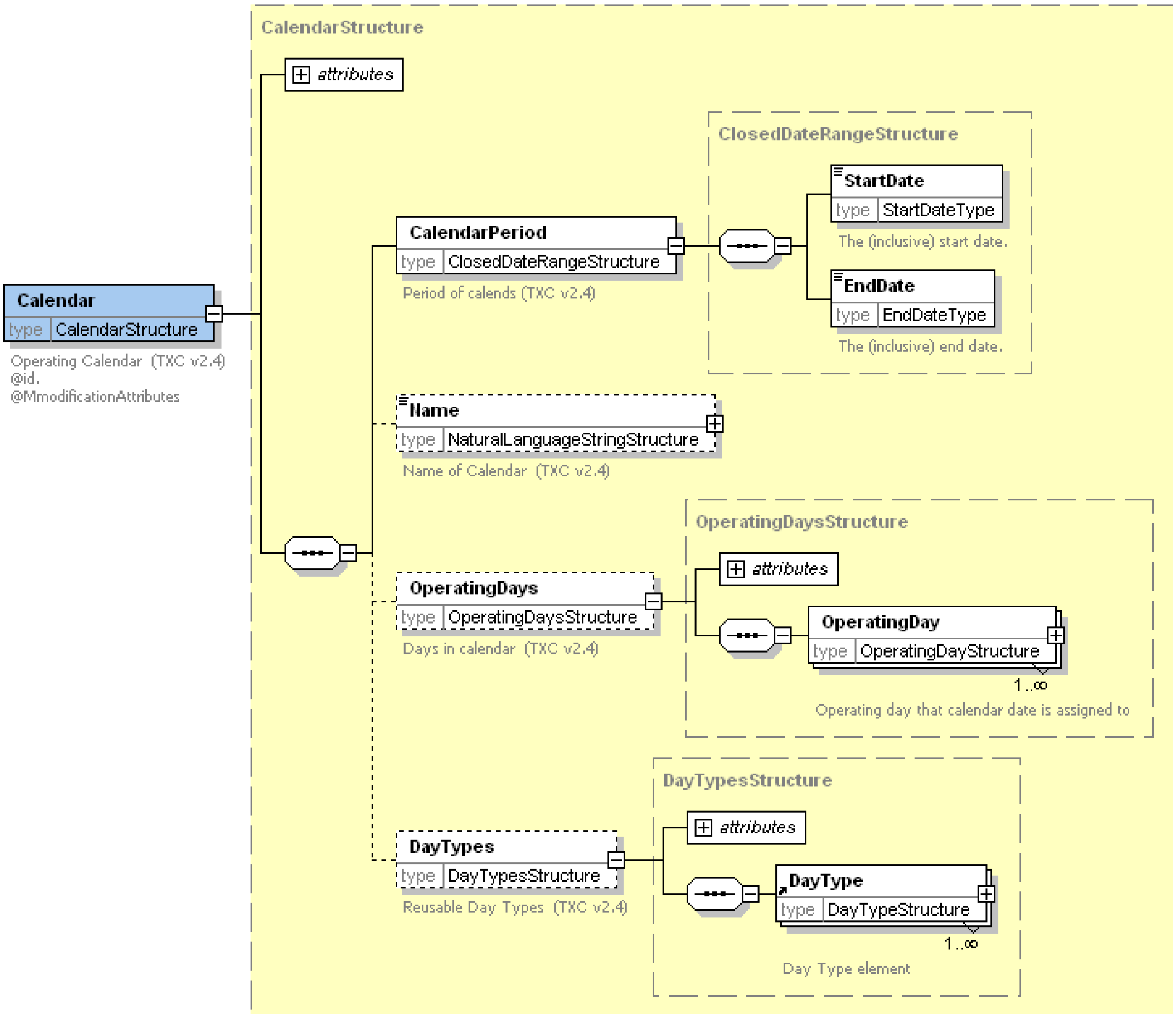


Figure 6-137 – Calendar Element

6.13.1.1 Calendar / OperatingDay Element (TXC v2.4)

The ***Calendar*** / ***OperatingDay*** element (Figure 6-138**)** specifies an actual calendar day and assignments it to a day type using a ***DayAssignment***. See Modelling operation days for precedence of overlapping dates.

- ***Date***: Calendar Date of Operating day.
- ***StartTime***: The (inclusive) start time. If omitted, the day is assumed to start at 00:00.
- ***Duration***: The length of the day. If omitted the day is assumed to be 24 hours long.
- ***Day Assignment***: Assignment of Day to an Individual day type.
- Day Identifier: specified either as
  - ***BankHolidayName:*** name of a predefined Bank Holiday. (See BankHolidays in 6.11.3
  - ***PublicHoliday:*** name of an arbitrary Public Holiday.
- ***DayTypeRef: Day type*** to which to assign ***OperatingDay***

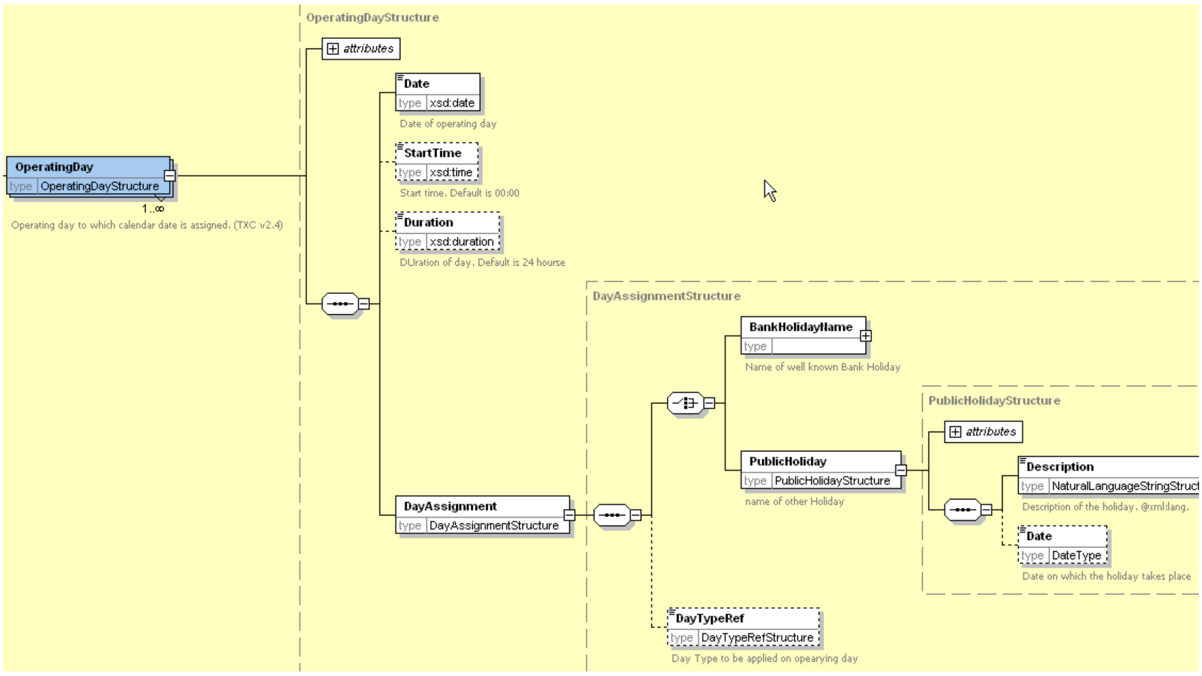


Figure 6-138 – Calendar / OperatingDay Element

6.13.1.2 Calendar / DayType Element (TXC v2.4)

The ***Calendar*** / ***DayType*** element (Figure 6-138**)** specifies an actual day type. It reuses the same property of day elements of an OperatingProfile element. See earlier

- ***Id***: Calendar Date of Operating day.
- ***NormalDays***: See ***OperatingProfile*** earlier.
- ***SpecialDays***: See ***OperatingProfile*** earlier.

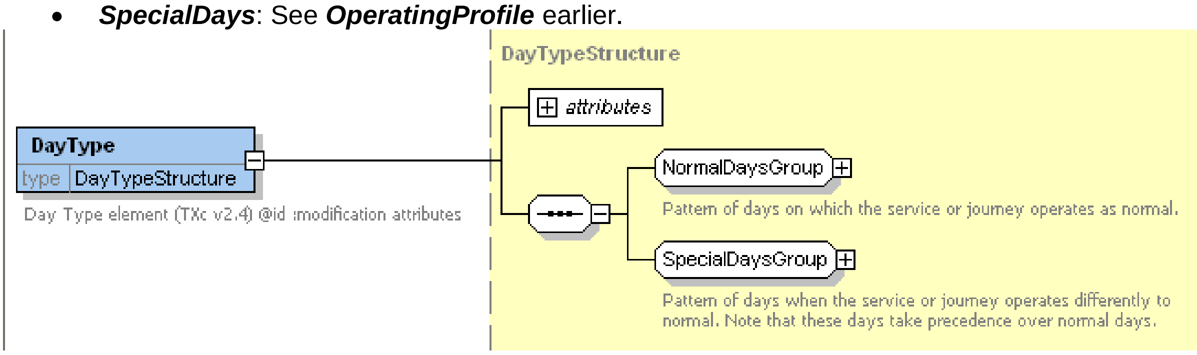


Figure 6-139 – Calendar / DayType Element

6.14 Miscellaneous Elements

6.14.1 SupportingDocument Element

The ***SupportingDocument*** element (Figure *6-140*) Associates any supporting documents associated with the whole *TransXChange* schedule document – other documents, for example a schematic map, may be associated with individual elements using specific tags. Documents may be in any file format and are identified by a ***DocumentUri***. Note that documents can also be associated more specifically with an individual ***Registration***. ***Registration*** / ***SupportingDocument*** should be used in preference to this element for documents associated with a ***Registration***

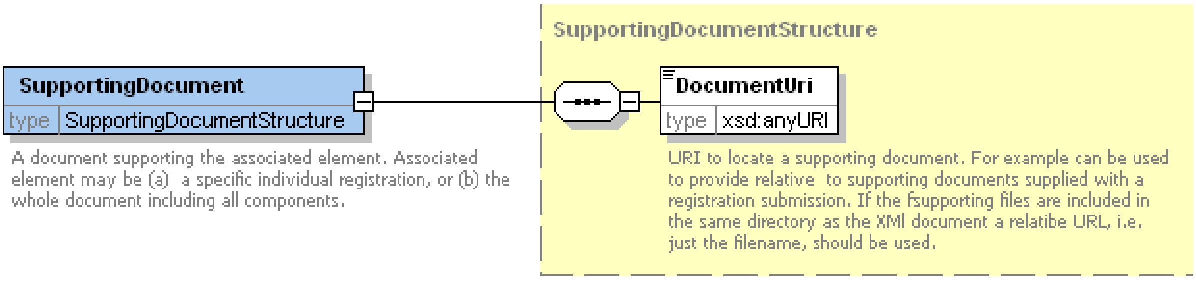


Figure 6-140 – SupportingDocument Element

7 COMMON SCHEMA ELEMENTS

Some elements and types are common to a number of different elements in the *TransXChange & NaPTAN* schemas. These are described here.

7.1 LocationStructure

The ***LocationStructure*** type (*Figure 7-1*) is used to describe the spatial position of a stop or other point, for example on a ***Location*** element. Coordinates may be specified in Grid or WGS84 formats, or both. The primary coordinates used can be indicated by the ***LocationSystem*** value (*Grid* or *WGS84*) specified on ***TransXChange*** document root elements. Coordinates must be supplied for all elements in the specified primary coordinates, and may optionally be provided in the other system as well. *NaPTAN* data should be submitted in *Grid* format. *NaPTAN* data will normally be distributed in both formats.
If *Grid* coordinates are provided:

- ***GridType:*** Nominated grid system e.g. UKOS , ITM, IrishOS; UKOS is assumed by default.
- ***Easting***: Easting grid coordinates of stop.
- ***Northing***: Northing grid coordinates of stop.

If *WGS84* coordinates are provided:

- ***Longitude***: Longitude of stop in *WGS84* coordinates.
- ***Latitude***: Latitude of stop in *WGS84* coordinates.

If Both Grid & *WGS84* coordinates are provided:

- ***Translation***, containing both of the above coordinate groups.

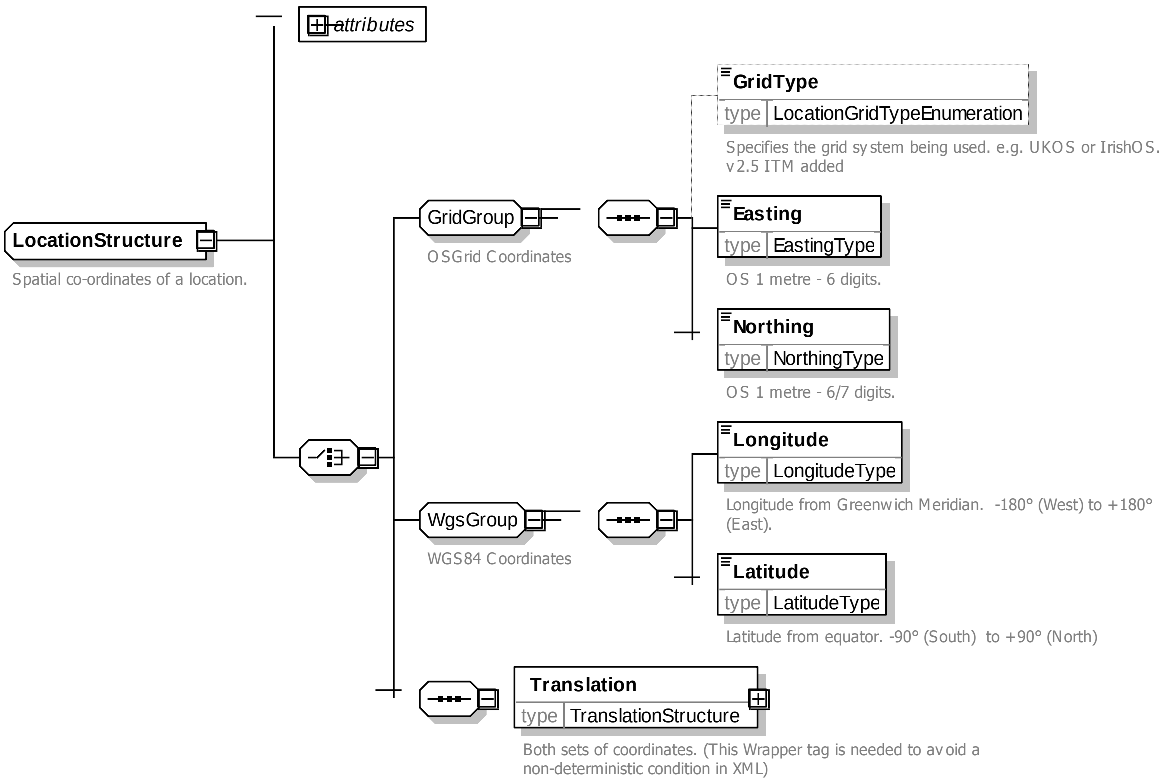


Figure 7-1 – LocationStructure

7.2 Duration Simple Type

The ***Duration*** simple type is used by a number of elements to specify a relative time in minutes and seconds. It uses a standard W3C ***duration*** type.
See http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/#duration.
Durations are encoded in the format *PT99M88S*, where *99* is the minutes and *88* is the seconds. For example, '*PT12M22S*' denotes twelve minutes and twenty-two seconds. The seconds may be

omitted for whole minutes, for example, *PT5M*. Note that the W3C format also allows years, month, week and day intervals as well but these are not needed for timing intervals. The W3C definition allows arbitrary integer values for the minutes and arbitrary decimal values for the number of seconds can include decimal digits to arbitrary precision. thus PT1201M, PT360.25S or PT1000S are valid (i.e. seconds do not have to be modulo sixty). Either seconds or minutes or both may be coded. Units may be combined in an arbitrary manner for example, P5M, PT300S and PT3M120S are all valid equivalent encodings of 5 minutes.

7.3 TelephoneContactStructure Element

The ***TelephoneContactStructure*** (*Figure 7-2*) element specifies a phone number:

- ***TelNationalNumber***: Full telephone number including STD prefix
- ***TelExtensionNumber***: Any extension number.
- ***TelCountryCode***: International country code for telephone. E.g. +44.

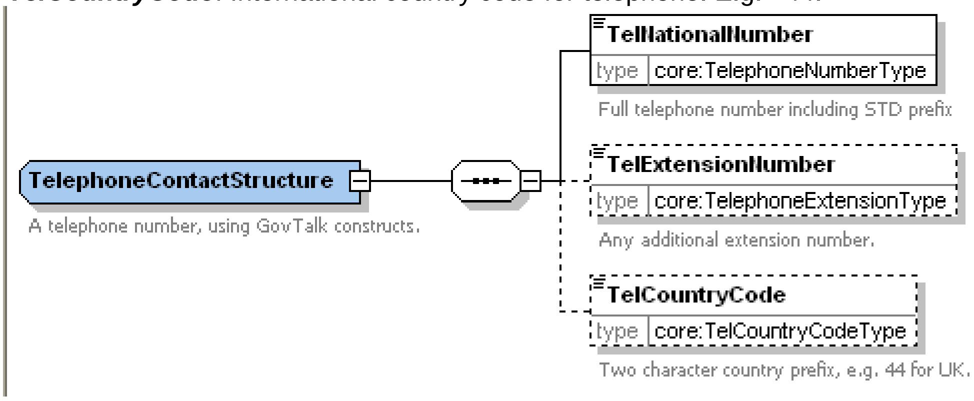


Figure 7-2 – TelephoneContactStructure

7.4 PostalAddressStructure Element

The ***PostalAddressStructure*** (*Figure 7-3*) element specifies a postal address. It is based on BS7666.

- ***Line***: Between two (minimum) and five (maximum) lines of address. Each line must be between one character and thirty five characters long. Characters are restricted to those shown in *Table 7-1*

- ***PostCode***: Post code of address. Valid UK Post code.

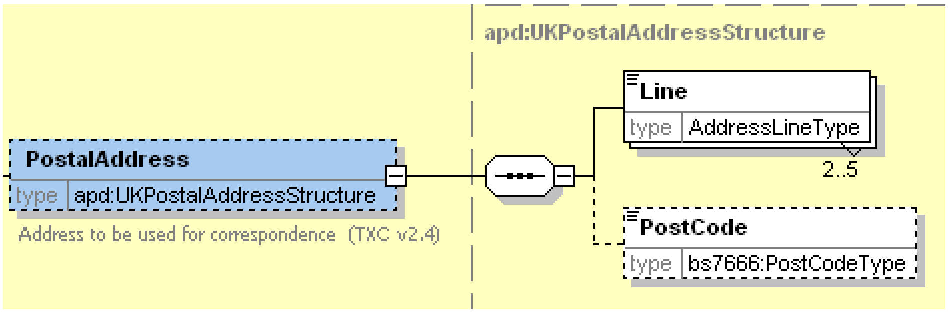


Figure 7-3 – PostalAddressStructure Element

Value	Description
a-z, A-Z	Upper or lower case alphabetic letters
0-9	Digits.
space	Space
“	Quote
‘	Apostrophe
,	Comma
.	Full stop
;	Semi-colon
:	Colon
*	Asterisk
+	Plus
-	Hyphen
_	Underscore
!	Exclamation mark
?	Question mark
=	Equals
<	Left Angle Bracket
>	Right Angle bracket
&	Ampersand
\	Back slash
/	Forward slash
(	Left Bracket
)	Right Bracket
{	Left curly Bracket
}	Right curly Bracket
[	Left square Bracket
]	Right Square Bracket
%	Percentage
#	Hash
@	AT
^	Caret
~	Tilde
$	Dollar
£	Pound
€	Euro

Table 7-1 – Allowed characters for Address Line

7.5 Note Element

A ***Note*** (*Figure 7-4*) models a set of notes attached to an element:

- ***NoteCode:*** Note identifier.
- ***NoteText:*** Text of note.

A ***Note*** may be used on ***Service, VehicleJourney & ConnectingVehilceJourney*** elements.

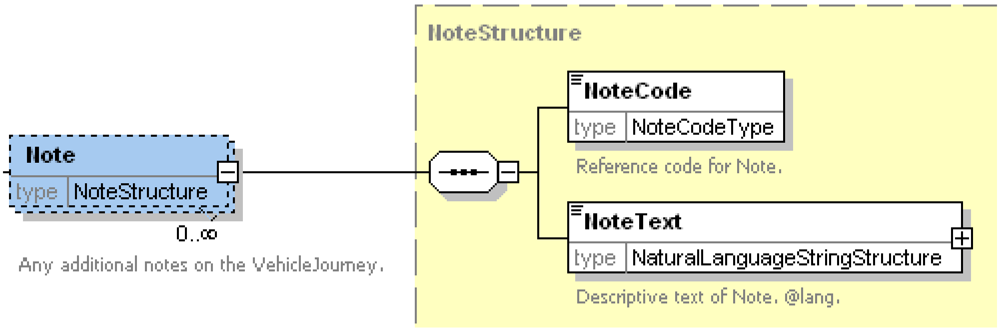


Figure 7-4 – Note Element

7.6 Submode Element

A Submode (*Figure 7-4*) describes a further classification of a mode: It must be one of nine values corresponding to the primary mode of the service. : ***AirSubmode, BusSubmode, CoachSubmode, FunicularSubmode, MetroSubmode, RailSubmode, TelecabineSubmode, TramSubmode, WaterSubmode.*** See JourneyWeb specification for allowed values for each enumeration.

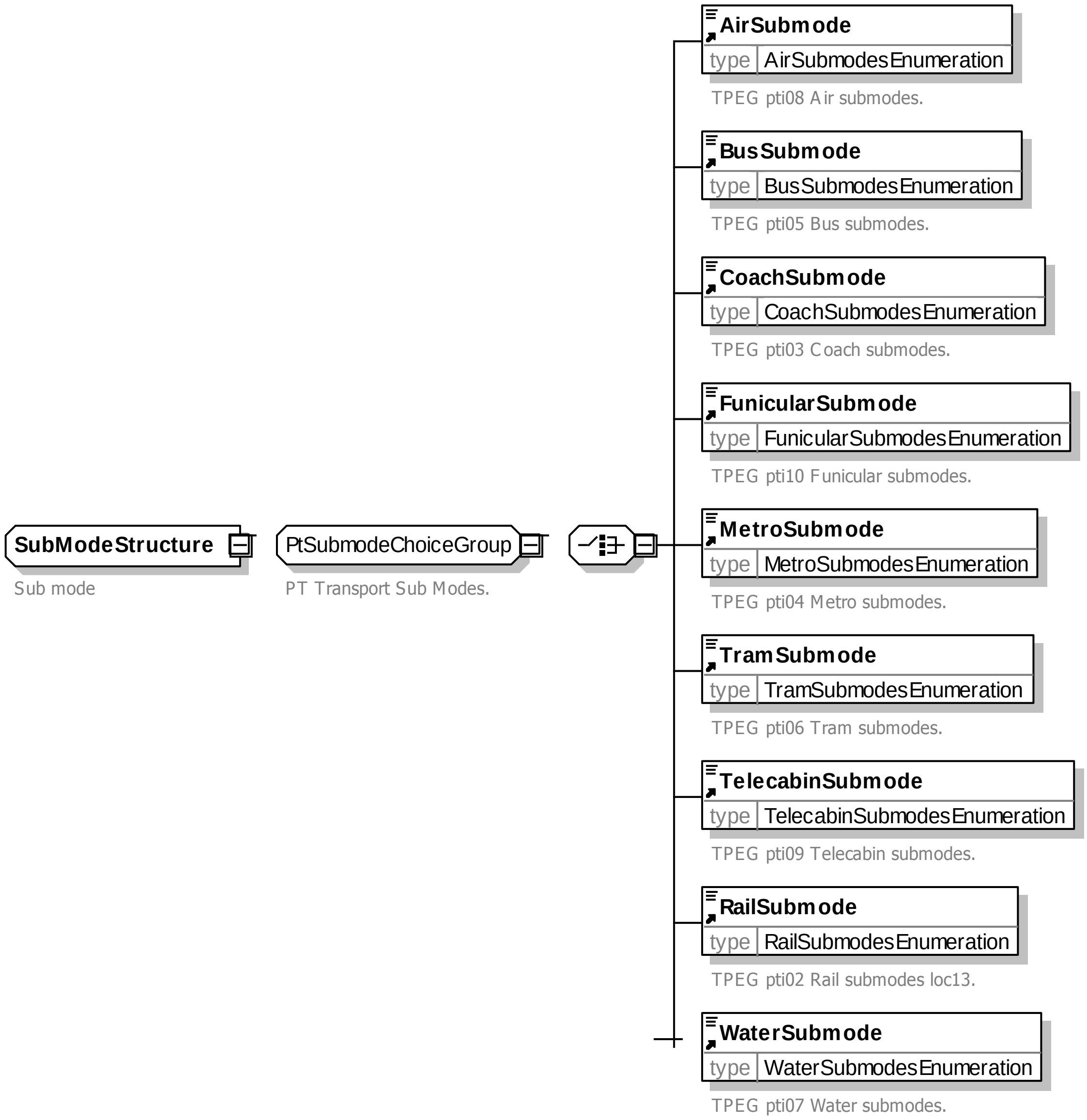


Figure 7-5 – Submode Element

8 ELECTRONIC BUS SERVICE REGISTRATION PROCESS

This section summarises the anticipated process for registering a Bus Service using *TransXChange*. The proposed process is subject to confirmation by VOSA following formal testing in a demonstration. Registration includes the following steps:

8.1 Step 1: Preparation

The Transport Operator creates a proposal for a bus service and follows his normal arrangements for consulting local authorities and others as appropriate before registering the proposal.

8.2 Step 2: Encoding

The Transport Operator or its agent transfers the proposal onto a computer system. This could be either a system that handles the scheduling of operations (and which includes the capability to output the registration as *TransXChange* registration compliant XML document), or a simpler system that only creates TransXChange registration files. Some operators may use an agency to do this work for them – and in some areas the local authority might offer to act as an agent, particularly in respect of contract services. Each service Registration will create a separate *TransXChange* file – and these will be referenced using the operator's next available registration number. Each change to a Registration likewise will carry a new sequential "version number".

8.3 Step 3: Transmission

The Operator or the Operator's agent logs onto the internet and connects to the VOSA Server with a normal web browser (MS Internet Explorer, Netscape, etc) using a previously-allocated username and password. The VOSA system provides a secure web connection over which the electronic registration details can be sent to the relevant Traffic Area Office. The VOSA service will offer a web page through which *TransXChange* files can be submitted, individually or in bulk. Files can be zipped (compressed) to reduce connection times – and multiple files can be submitted in a single zipped file. Files will be stored in a secure area of the VOSA web site and will be accessible only to the relevant Traffic Area staff, the operator making the submission and to the local authorities in whose area the service is to operate.

8.4 Step 4: Validation

The VOSA system will check that each file (unzipped, if necessary) meets the technical requirements of *TransXChange* – and will send a message back to the operator immediately if the file(s) fail this test. If each file passes the test, then the VOSA system will send an e-mail the relevant local authority (or authorities) and to the operator to advise them that a registration(s) has been submitted and they can collect the submitted file(s) through their own internet connection to a secure area of the VOSA web site

8.5 Step 5: TAN Review

Copies of the submitted file(s) are now passed into the Traffic Area Office's business system for review in the appropriate Traffic Area Office. Some automatic checks are made on the content of the file – and the report of these checks is then passed to a case worker who will review the proposal and, once any problems are resolved, issue the acceptance for each Registration. The acceptance creates a new file – in PDF format – which provides an unchangeable record of the "registered particulars" contained in the *TransXChange* file. This file will be put into the secure area of the VOSA web site. Both the operator of the service, and the local authorities in whose area the service is to operate, will be advised by e-mail that the Registration has been accepted and that the PDF file of its registered particulars is available for downloading securely from the web site. If problems are found with the registration proposals during this process, the operator may be invited to make changes to their proposals and to resubmit them, starting again at Step 2 of this process.

8.6 Step 6: Acceptance and Distribution

The operator who submitted the registration (and the relevant local authorities) can then download a copy of the PDF file and can view the content of this file using freely available software (such as Adobe Acrobat Reader). This provides confirmation of the acceptance of the Registration – and sets out the only information (particularly the timetable shown only at principal “timing” points) to which the Traffic Commissioner can refer in any enforcement proceedings.

The files submitted or created during this process will remain accessible through the VOSA web site for up to 90 days using the secure access codes provided in the e-mails sent to the operator and to the relevant local authorities. After that period, the files can still be obtained on request from the relevant Traffic Area Office.

The electronic Registration process will be the same, whether the proposal is to register a new service, to change an existing registered service, or to cancel an existing service. Changes to an existing Registration require the re-submission of the complete registration details using *TransXChange* (but most of these details will have been stored in the operator’s systems for re-use in such circumstances). Cancellations require the submission of a very small TransXChange file that identifies the Registration concerned and the last date of operation.

TransXChange files can include timetables for use not only on normal operating days, but also those which will be used on Bank Holidays and on other special days (such as those around Christmas and the New Year). Operators will be encouraged to make full use of these facilities so that special timetables are available for public information systems well in advance of each special day of operation, and to avoid the need to submit special registrations (or notifications) for such services.

9 THE TRANSXCHANGE PUBLISHER

The *TransXChange* Publisher is a free tool issued along with *TransXChange,* which allows users to render *TransXChange* XML documents into a readable timetable-like layout, that uses the *Acrobat* pdf file format. See Figure 9-1. The free **Acrobat** reader from Adobe Inc. (http://www.adobe.com) can be used to read and print pdf files.

The Publisher can be invoked from a Desktop GUI. It has options to produce

- **Particulars**. The particulars section includes a summary of the contents of the TransXChange document, (for example how many stops and journeys) followed by a textual listing of the entities described in the file (such as operators, services, routes, and stops).
- **Timetable**. The timetable section contains matrix timetables for the services in the *TransXChange* document. Separate timetables are generated for different services, directions (e.g. outbound and inbound), and day types (e.g. a Monday to Friday timetable, and a Saturday timetable).
- **Diagnostic Report.** The diagnostics section contains a report detailing violations of consistency checks for the TransXChange document (over and above those expressed in the TransXChange XML Schemas alone).
- **Route Track**. The route track section is a separate pdf document. It consists of route plots for the services in the TransXChange on a map background along with an accompanying table of stops. It requires an on-lien connection to use.

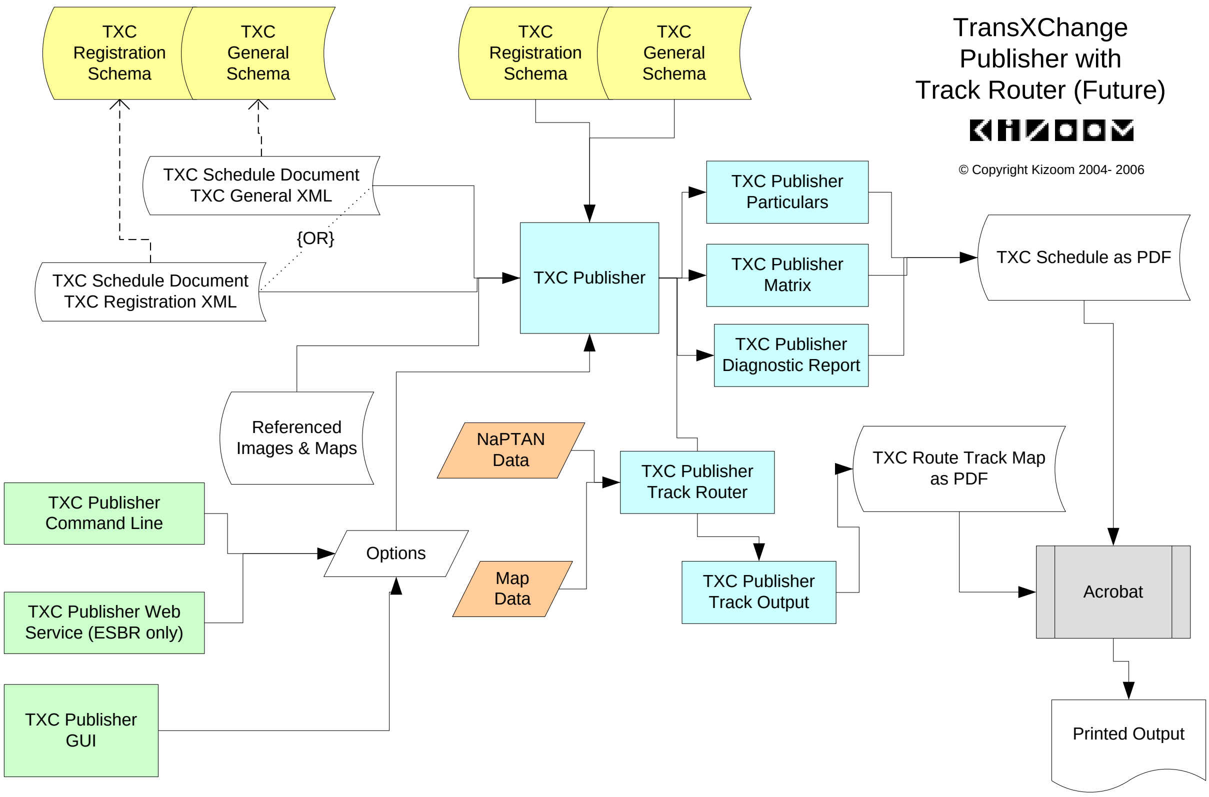


Figure 9-1 – Publisher

9.1 Required Environment

The *TransXChange Publisher* requires the installation of a standard open source environment for running Java (Java Runtime Environment 1.4.2 or higher). See Installation instructions for platform requirements.

The Route Track option requires a broadband internet connection to access the web services that provide stop and map data.

9.2 Installation Process

The Publisher is available as a downloadable zip at http://www.transxchange.org.uk.

Installation instructions and examples are included on the site.

9.3 Run Time Options

The Publisher has a number of run time options
(a) To control the content to be included.
(b) To specify various aspects of the rendering of content.

9.4 Generalised list of Publisher parameters

Group	*Parameter*	*Data type*	*Default*	*Description*	*WS*	*Comm and Line*	*GUI*
Input Operands	**Document-Path**	*url*	Required	*Name and Path to TransXChange XML document and associated files that are to be published.*	*Y*	*Y*	*Y*
Output Operands	**OutputPath**	*url*	Optional	*Output directory in which to place published output. If not otherwise specified, output is placed in the same directory as the input document*	*Y*	*Y*	*Y*
Processing options	**ValidateXML**	*boolean*	true	*Apply XML validation -*	*Y*	*Y+*	*Y*
Output Section Content Options	**Auto**	*Vosa |vosaAll |full*		*Options controlling the interpretation of auto* *See Parameter defaults below -*	*Y*	*Y*	*Y*
	Particulars	*none | basic | full*	full	*Include the particulars in output.* • *auto – default by pub format* • *none – no particulars* • *basic –basic particulars*	*Y*	*Y [1]*	*Y*
	Timetable	*none | basic | full*	full	*Include the timetable matrix in output.* • *auto – default by pub format* • *none – no matrix* • *basic – Omit footnotes* • *full - Include the timetable footnotes in output.*	*Y*	*Y [1]*	*Y*
	RouteTrack	*none | plain | basic | tiled*	none	*Include the route track in output. Default is false.* • *auto – default by pub format* • *none – no route track* • *plain – no map tiles* • *basic – Omit stop list*	*Y*	*Y+*	*Y*

	Embed	*boolean*	true	*Include any embedded image content in output.*	*Y*	*N*	*Y*
	Diagnostics	*auto | none | full*	auto	*Publish a diagnostic section.*	*Y*	*Y [2]*	*Y*
Filters	**TimingPoints**	*All |PTP |*	All [3]	*Include timing points of this type.*	*Y*	*Y [3]*	*Y*
	MergeFrequencyMergeFrequency-Journeys	*boolean*	true	*Merge similar frequent journeys into a single column.*	*Y*	*N*	*Y*
Route Track Map	**RouteScale**	*Auto | Small | Large*	Auto	Scale to use when tiling map. Small: 1:10,000, Large 1:50:000. Auto: scale to size	*Y*	*N+*	*Y*
	Route-Grouping	*Single | ByDirection*	false	*One route per map, or per direction.*	*Y*	*Y+*	*Y*
	RouteTiling	*A4 | none*	A4	*Output as A4 tiles or single image.*	*Y*	*N+*	*Y*
	StopData	*localOnly | web service*	Web-Service	*Source of stop coordinates.*	*Y*	*N*	*Y*
	MapData	*none | web service*	Web-Service	*Source of map tiles. Only used if RouteTrackMap specified.*	*Y*	*N*	*Y*
Watermark	**Background**	*Official | Vosa | Other*	Vosa	*Controls image.*	*Y*	*[N]*	*N*
Watermark	**Rubric**	*Official | Vosa | Other*	Vosa	*Controls headings and watermark.*	*Y*	*[N]*	*N*
Rendering	**Output-Format**	*Pdf | html*	pdf	*Output format pdf*	*Y*	*(Y) [4]s*	*Y*

Table 8-9 – Publisher Interface Parameters

[1] Command line by suppressing other parts: timetableOnly.
[2] Controlled in command line by suppression: novalidation.
full
[4] Matrix only/ HTML output is a Debug Tool -

9.5 Publishing Actions

The Publisher publishes a document in the following order:

9.5.1 Particulars & Service Matrix

- Summary Page
- Operator
- Serviced Organisations
- Services
- Registrations
    - i. ShortNoticeRegistrations
- Lines
- Routes
    - i. Local Stop Declarations
    - ii. References to existing stops
    - iii. Embedded Map
- Fixed Route Services
    - i. Outbound VehicleJourneys
        1. Monday to Friday Built-in Journey Grouping
            a. Matrix
            b. Notes
        2. Saturday Built-in Journey Grouping
            a. Matrix

            b. Notes
        3. Sunday Built-in Journey Grouping
            a. Matrix
            b. Notes
        4. Custom Journey Groupings
            a. Matrix
            b. Notes
    ii. Inbound VehicleJourneys
        1. Monday to Friday Built-in Journey Grouping
            a. Matrix
            b. Notes
        2. Saturday Built-in Journey Grouping
            a. Matrix
            b. Notes
        3. Sunday Built-in Journey Grouping
            a. Matrix
            b. Notes
        4. Custom Journey Groupings
            a. Matrix
            b. Notes
- Flexible Route Services
    i. Flexible Stops
    ii. Fixed Stops
    iii. Timebands
- Supporting documents

9.5.2 Route Map

- Outbound
    i. Service Map
    ii. List of Outbound route Stops
- Inbound
    i. Service Map
    ii. List of inbound Route Stops

10 NAMING & CODING CONVENTIONS

Systematic naming conventions and a consistent coding style are used in the *TransXChange* schemas. These conventions are summarised in this section.

10.1 Naming of Elements

TransXChange follows consistent principle for naming elements:

10.1.1 Use of Camel Case

Camel case is used for all names in the XML schema:

- Upper camel case is used for element and attribute names, for example ***JourneyPatternTimingLink***, ***HailAndRide***.
- Lower case is however used for two standard attributes: ***xsd:lang*** and ***id***, following W3C usage.
- Lower camel case is preferred or enumerated character values, for example '*saturdayMorning',* except for proper names, which may be capitalised, e.g. '*IsleOfMan*
- Acronyms are treated as words for capitalisation, thus ***TanCode***, not ***TANCode***. This is one point where we follow common best practice but diverge from e-gif. Treating acronyms as words allows for a uniform parsing of names to derive their components, and avoids ambiguity on case of contiguous acronyms, for example ***TANAPD*** vs. ***TanApd***, or one letter words contiguous with an acronym, for example ***DialATAN*** vs. ***DialATan.***

10.1.2 Use of Standard Name Suffixes

TransXChange and *NaPT schema e*lement, type and attribute names have been revised along consistent principles:

- All simple types end with the suffix '***Type'***.
- All complex types end with '***Structure'***.
- All enumerations end with '***Enumeration'***.
- All groups end with '***Group'***.
- Elements representing references to other entities are suffixed with '***Ref'***.
- Externally referenced identifiers of entities are generally suffixed with '***Code'*** (and represented as elements). Code values are usually unique for the element type within a document.
- Internally referenced identifiers are generally named with '***id'*** (and represented as attributes). id attributes typically have a keyref constraint on their uniqueness. The uniqueness scope for id attributes is normally for the element type within an instance document, but could also be just within an instance of specified element.
- Externally referenced classifiers of entities are generally suffixed with '***Classification'*** (rather than say '***Type'***). (Some exceptions are made to this rule for legacy usage).
- Externally referenced names of entities are generally suffixed with '***Name'***. If the context is readily apparent they may be called just Name.
- Natural Language text descriptions of entities are generally termed '***Description'***.

10.1.3 Meaningful Names

Several other consistent naming principles are followed:

- Abbreviations are generally avoided – for example '***Operations'*** is preferred to 'Op'.
- A container element representing a one-to-many relationship is in the plural; for example, ***StopPoints*** contains one or more ***StopPoint*** elements.
- We avoid repeating the name of the parent element as an adjective in individual child elements, except where for semantically important elements. Thus for example, ***Author*** contains ***Title***, ***Position***, ***Forename***, ***Surname***, not ***AuthorTitle***, ***AuthorPosition***, ***AuthorName***, ***AuthorSurname***
- We avoid the use in domain elements names of terms that have strong software connotations:

- The suffixes '***Type'*** and '***Group'*** are avoided in element names except for internal schema elements.
- The term '***Exclusion'*** is used generically to denote an exclusion period for the service (rather than the previous term Exception) e.g. ***JourneyPatternExclusion***.

10.1.4 Semantically Significant Order

Several principles are used to order the subelements at any given level of containment:

- When declaring elements within a parent, subelements are placed in a consistent general order according to the nature of their role as follows:
  - (i) Elements that identify the entity, such as codes or numbers.
  - (ii) Elements that classify the entity.
  - (iii) Elements that describe the element in text, such as names or descriptions.
  - (iv) Elements describing other properties of the entity.
- Where there is an inherent temporal order, elements are placed in temporal sequence.

10.1.5 Standardised Terminology

An attempt has been made to use the appropriate *Transmodel* term wherever appropriate. For example ***Garage*** rather than ***Depot***. The main divergences from *Transmodel* are listed in section 13.2.

10.2 Typing of Elements

Some general principles are used for typing values.

- Explicit, specific types are used wherever possible, for example ***Duration***:
- Complex types are declared for all significant elements.
- Internally referenced identifiers are generally of type IdType, which is defined to be of base XML type NMTOKEN.
- Elements whose content is a text string in a national language are of type ***NaturalLanguageStringStructure.***

10.3 Element Constraints

Some general principles are used for constraining values.

- *Mandatory Elements are normally populated.* XML constraints are usually specified to ensure mandatory elements are populated, for example strings should contain at least one character.
- *Optional elements not empty:* Where alternative structures are available, the absence of an element is not relied upon to infer meaning. Instead an empty element or attribute value is used to make the condition explicit, or there is a default value defined. This principle has been generally been followed for new and remodelled features.

10.4 Use of Attributes

In *TransXChange,* XML element attributes are generally used only for metadata, that is, data about data, such as change dates, or internal identifiers. *Table 10-1* summarises the attributes used in *TransXChange.*

Group	Element	Attribute	
Document Version	***TransXChange*** root element.	***CreationDateTime***	*1.2*
		ModificationDateTime	*1.2*
		FileName	*2.0*
		SchemaVersion	*1.2*
		Modification,	*1.2*
		RevisionNumber	*1.2*
		DataSource	*2/4*
Entity Version	***StopPoint, StopArea, NptgLocality, Service, VehicleJourney, FlexibleVehicleJourney, Route, RouteLink, FlexibleZone, Registration, JourneyPattern, Operator, JourneyPatternInterchange,***	***CreationDateTime***	*2.0*
		ModificationDateTime	*2.0*
		Modification,	*2.0*
		RevisionNumber	*2.0*

	JourneyPatternSection, ServicedOrganisation,	***Status,***	*2.0*
	JourneyPatternTimingLink, VehicleJourneyTimingLink, VehicleJourneyInterchange, PositioningLink, LayoverPoint, DeadRun, Garage, DayType, DayAssignment, AccessVehicleEquipment, ConnectingVehicleJourney, JourneyGrouping, DataRight, Contributor.	As above	*2.4*
Child Entity Version	***OperatorLicence, Descriptor, FlexibleServiceTimes, AnnotatedCrossRef, StopValidity, VariableStopAllocation***	As above	
id	***Route***	***id***	*1.2*
	JourneyPattern	***id***	*1.2*
	DeadRun	***id***	*2.0*
	RouteSection	***id***	*2.0*
	JourneyPatternSection	***id***	*2.0*
	RouteLink	***id***	*1.2*
	JourneyPatternTimingLink	***id***	*1.2*
	VehicleJourneyTimingLink	***id***	*1.2*
	PositioningLink	***id***	*2.0*
	JourneyPatternStopUsage	***id***	*1.2*
	VehicleJourneyTimingLink	***id***	*1.2*
	JourneyPatternInterchange	***id***	*1.2*
	VehicleJourneyInterchange	***id***	*1.2*
	JourneyGrouping (Custom & Built-in)		
	ConnectingVehicleJourney	***id***	*2.4*
	Calendar	***id***	*2.4*
	DayType	***id***	*2.4*
	OperatingDay	***id***	*2.4*
	DayAssignment	***id***	*2.4*
	Location	***id***	*1.2*
Data	***Location***	***Precision***	*1.2*
	JourneyPatternStopUsage, VehicleJourney	***SequenceNumber***	*2.0*
	Route / Track / MapSystemReference	***MappingSystem***	*2.0*
Language	Text elements***: Name, Description,*** etc. See section on National Language Support	***xml:lang***	*2.0*

Table 10-1 – TransXChange Attributes

10.5 Implementation of Model Relationships

In *TransXChange,* some stylistic conventions are used to make clear the mapping of the reference model relationships into the XML schema.

- Significant entities have a uniquely scoped identifier (always an element named ***xxxCode,*** or ***xxxNumber***, or an id attribute).
- Relationships are implemented by placing a reference to the identifier as a foreign key on the referencing element (shown by the navigability arrow in UML diagrams). The reference has the form ***xxxRef***. For example, ***StopPoint*** is identified by an ***AtcoCode,*** and referenced in relationships by a ***StopPointRef***.
- Container elements are generally used for significant one-to-many relationships, for example ***StopPoints*** contains the ***StopPoint*** elements. In the Delta schema can be marked as 'delta' to indicate that only a subset of elements are supported.

11 NATIONAL LANGUAGE SUPPORT

TransXChange is enabled to allow the coding of schemas in different National Languages, such as Welsh.

11.1 Text Content Types

The textual data of a *TransXChange* schedule falls into three different categories:

- **Fixed Text**: National Language Translations of fixed encoded *TransXChange* values (for example the TAN area names), and terminology for concepts such as 'Service' rendered when using a style sheet to transform a schedule into a published format.
- **Free Text:** The contents of data elements that can be specified as content for textual elements (having an xml:***lang*** attribute and a type of ***NaturalLanguageStringType***), for example operator names, route descriptions and other notes.
- **External Data:** The contents of data fetched from external data systems, for example *NaPTAN* stop names.

11.1.1 Use of Fixed Text

An overall **xml:lang** attribute is specified at the schema level on the ***TransXChange*** root element. This specifies the default language for the schedule, i.e. the default implied language that is to be used to publish the timetable. It defaults to English.

- Translations can be established for the text associated with the different fixed elements.

11.1.2 Use of Free Text

Elements which may contain free text in a natural language (*Table 11-1*), such as Welsh or English, have an **xml:lang** language attribute to indicate the language in which they are.

- English is assumed if no attribute is specified.
- The provision of alternative names for a stop in different languages is covered by *NaPTAN*, which allows for multiple alternative names.

Group	Element	Note	
StopPoint	***CommonName***	Use NaPTAN	2.1
	NptgLocalityName		2.1
	StopPoint / Short CommonName		2.1
	StopPoint / Landmark		2.1
	StopPoint / Street		2.1
	StopPoint / Crossing		2.1
	StopPoint / Indicator		2.1
	StopPoint / Town		2.1
	StopPoint / Suburb		2.1
	StopPoint / Note		2.1
StopArea	***StopArea / Name***		2.1
Organisation	***ServicedOrganisation / Name***		2.1
	DatePattern / Description		2.1
FlexibleZone	***FlexibleZone / Description***		2.1
Route	***Route / Description***		2.1
	Route / Manoeuvre		2.1
	Instruction / Summary		2.1
	Track / Feature / OnwardName		2.1
	Track / Feature / Description		2.1
Service	***Origin***		2.1
	Destination		2.1
	Vias /ViaName		2.4
	ServiceClassification / OtherService / Description		2.1
	StopRequirements / NewStopsRequired / Note		2.1
	Description		2.1
	Note / NoteText		2.1
	ToBeMarketedWith / RelatedService / Description		2.1
	ContractedService / Description		2.1
	QualityPartnership		2.1

JourneyGrouping	***Description***		2.4
	Vias /ViaName		2.4
JourneyPattern	***DestinationDisplay***		2.1
	Description		2.1
	Vias /ViaName		2.4
	Block / Description		2.1
	Block / Note		2.1
	LayoverPoint / Name		2.1
	VehicleType / Description		2.1
	Frequency / Description		2.1
	JourneyPatternTimingLink / StoppingArrangements		2.1
	JourneyPatternTimingLink / StopUsage / Note		2.1
	JourneyPatternTimingLink / DynamicDestinationDisplay		2.4
Flexible-JourneyPattern	***Description***		2.1
VehicleJourney	***Note***		2.1
Interchange	***Interchange / StoppingArrangements***		2.1
OperatingProfile	***OtherPublicHoliday / Description***		2.1
Operator	***OperatorNameOnLicence***		2.1
	Garage / GarageName		2.1
Registration	***Author / Position***		2.1
	SubsidyDetails / SubsidisingAuthority		2.1
	StopRequired		2.1
	NoteText		2.1
	QualityParnership		2.1
Short Notice Registration	***PublicAvailability / NonAvailabilityDescription***		2.1
	ChangeImpact / MinorChangeDescription		2.1
	ReplaceDiscontinuedService / DiscontinuedService/ Description		2.1
	LocalHolidayChange / LocalHolidayNote		2.1
	SpecialOccasion / SpecialOccasionName		2.1
	RegulationOrderCompliance / TrafficOrderNote		2.1
	OtherServiceType / Description		2.1
	ChangeRequestedByExternalAuthority / ChangeRequestDescription		2.1
	MiscellaneousJustification		2.1
DataRight	***Name***		2.4
	TermsAndConditions		2.4
	CopyrightStatement		2.4
	PolicyJustification		2.4
Calendar	***Name***		2.4
Line	***LineName***		2.1
DeadRun	***Name***		2.1
VehicleEquipment	***Description***		2.4

Table 11-1 – Elements That May Contain Natural Language Text

11.1.3 External Data

Any national language alternatives of **StopPoint** and **StopArea** names are provided by *NaPTAN*. The schema xsd:lang attribute should be used to determine the preferred language alternative to use when rendering names in timetables.

11.2 Publishing or Exchanging Documents

Note that the free text elements may only be in one language at a time in a given document. In order for the language specific free text elements of a schedule to be exchanged in multiple languages, the schedule must be republished in each language in turn.

12 VERSIONING

TransXChange schemas and documents must be versioned with an explicit version number so as to manage change in a distributed operating environment, and in particular to allow the inter-operability of versions of *TransXChange* running concurrently on different systems.

12.1 Version Numbering Convention

TransXChange follows the e-Gif convention for version numbering.

- *Released* Version numbers have the form *n.m*, (e.g. '3.0').
- *Drafts* have the form *n.mx* (e.g. '3.1a').
- The main version number (*n*) will be incremented when the change from the previous version of the schema will cause existing documents to fail to validate. For example if a new mandatory element is added.
- The minor version number (*m*) will be incremented when the change to the schema will allow existing documents to continue to validate. However some new documents may fail to validate against the old version (for example, if a new optional element is added).
- The draft version number (*x*) indicates that the version is still under discussion and may be subject to further changes. Generally it will be incremented to indicate a material change to a previous release or previous draft. Intermediate drafts will usually be withdrawn once they are superseded.

12.2 Resource Versions

12.2.1 Schema URI Version

In line with W3C practice, a separate directory and URL will be used for each version of the schemas; the schema name will remain the same (N.B. a directory rather than document level numbering system is preferred for the leaf schemas because it facilitates the management of multiple components of a modularised schema, and multiple document artefacts).
For example

- http://www.transxchange.org.uk/schemas/2.0f/TransXChange_registration.xsd

and

- http://www.transxchange.org.uk/schemas/3.1/TransXChange_general.xsd

Different versions will coexist at the same time. Old versions will generally first be deprecated, and then retired.

12.2.2 Namespace URI Version

e-GIF mandates that Namespace URI should not be versioned. (A different URL for the namespace and the schema) The following URI will be used for namespace.

- http://www.transxchange.org.uk/schemas/

12.2.3 Package Versions

TransXChange embeds a number of common type definition packages that are shared with other UK standards. For convenience, a separate copy of the common packages is distributed with each standard. The individual package files are given version numbers in line with the e-GIF system in order to ensure the correct version is used.
For example, for the shared NaPT stop definition types file might be called *NaPT_stop-v1-0.xsd.* It will be distributed in *TransXChange* as:

- http://www.transxchange.org.uk/schemas/2.0/napt/NaPT_stop-v1-0.xsd

12.3 Packages

The *TransXChange* model is modularised into a number of packages, with a strict linear dependency. See Figure 12-1 & Figure 12-2.

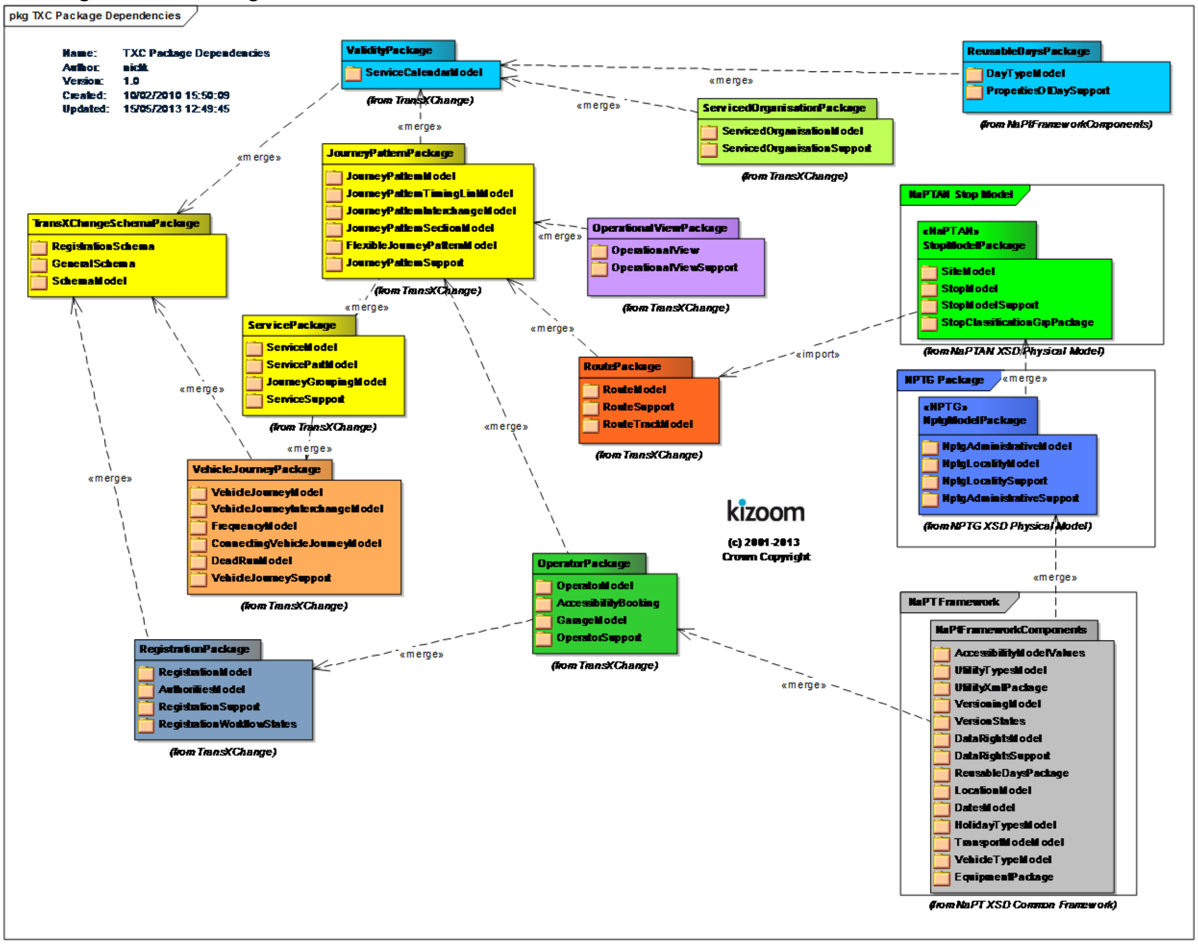


Figure 12-1 – TransXChange Packages

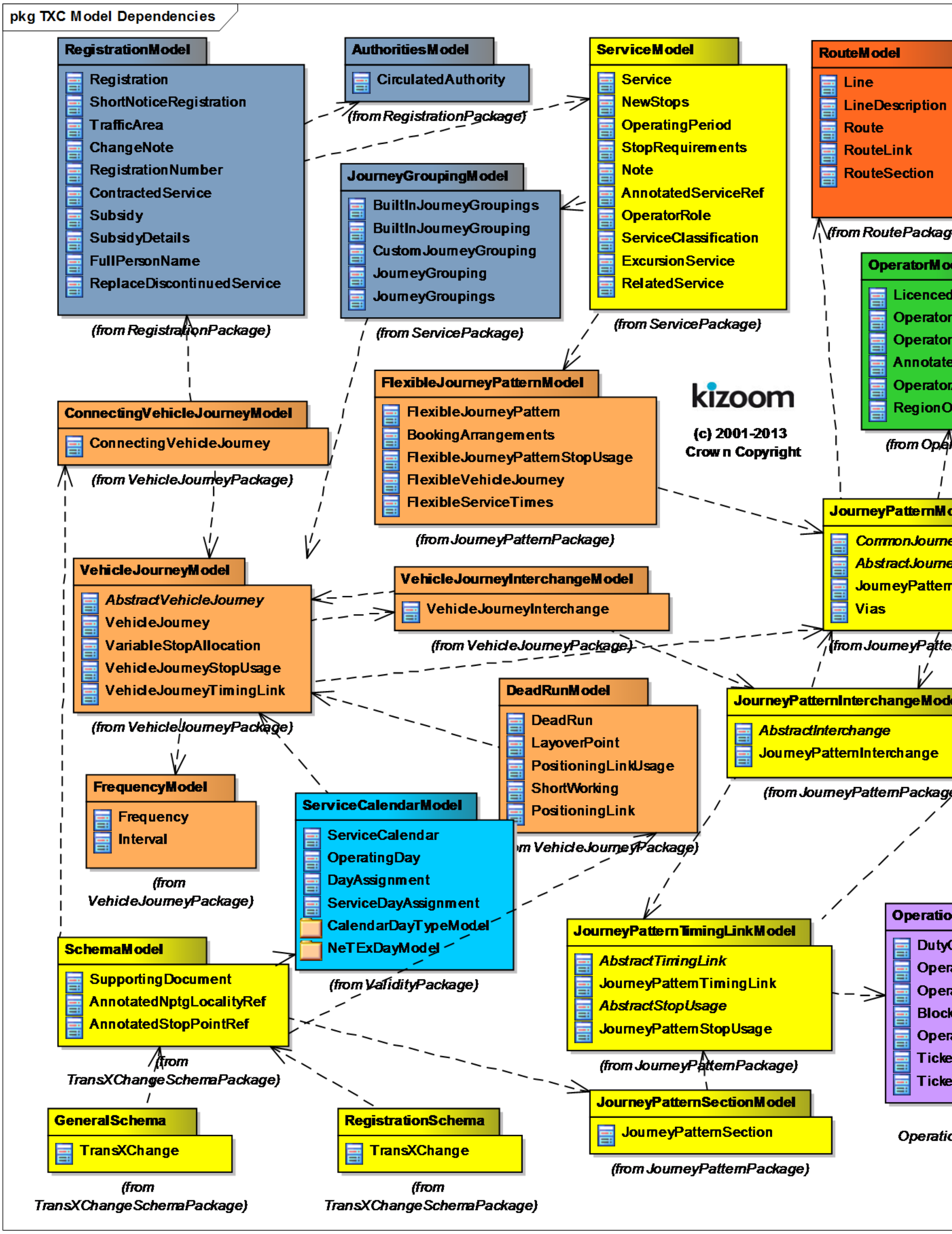

pkg TXC Model Dependencies
RegistrationModel
Registration
ShortNoticeRegistration
TrafficArea
ChangeNote
RegistrationNumber
ContractedService
Subsidy
SubsidyDetails
FullPersonName
ReplaceDiscontinuedService
(from RegistrationPackage)
AuthoritiesModel
CirculatedAuthority
(from RegistrationPackage)
ServiceModel
Service
NewStops
OperatingPeriod
StopRequirements
Note
AnnotatedServiceRef
OperatorRole
ServiceClassification
ExcursionService
RelatedService
(from ServicePackage)
RouteModel
Line
LineDescription
Route
RouteLink
RouteSection
JourneyGroupingModel
BuiltInJourneyGroupings
BuiltInJourneyGrouping
CustomJourneyGrouping
JourneyGrouping
JourneyGroupings
(from ServicePackage)
FlexibleJourneyPatternModel
FlexibleJourneyPattern
BookingArrangements
FlexibleJourneyPatternStopUsage
FlexibleVehicleJourney
FlexibleServiceTimes
(from JourneyPatternPackage)
kizoom
(c) 2001-2013
Crown Copyright
ConnectingVehicleJourneyModel
ConnectingVehicleJourney
(from VehicleJourneyPackage)
VehicleJourneyModel
AbstractVehicleJourney
VehicleJourney
VariableStopAllocation
VehicleJourneyStopUsage
VehicleJourneyTimingLink
(from VehicleJourneyPackage)
VehicleJourneyInterchangeModel
VehicleJourneyInterchange
(from VehicleJourneyPackage)
DeadRunModel
DeadRun
LayoverPoint
PositioningLinkUsage
ShortWorking
PositioningLink
FrequencyModel
Frequency
Interval
(from VehicleJourneyPackage)
ServiceCalendarModel
ServiceCalendar
OperatingDay
DayAssignment
ServiceDayAssignment
CalendarDayTypeModel
NeTExDayModel
(from ValidityPackage)
JourneyPatternInterchangeModel
AbstractInterchange
JourneyPatternInterchange
SchemaModel
SupportingDocument
AnnotatedNptgLocalityRef
AnnotatedStopPointRef
(from TransXChangeSchemaPackage)
JourneyPatternTimingLinkModel
AbstractTimingLink
JourneyPatternTimingLink
AbstractStopUsage
JourneyPatternStopUsage
(from JourneyPatternPackage)
JourneyPatternSectionModel
JourneyPatternSection
(from JourneyPatternPackage)
GeneralSchema
TransXChange
(from TransXChangeSchemaPackage)
RegistrationSchema
TransXChange
(from TransXChangeSchemaPackage)

Figure 12-2 – TransXChange Model Dependencies

The XML schemas are organised according to corresponding package groups (*see Table 12-1*). *TransXChange* schemas are placed in the root folder, prerequisite shared schemas are placed in subfolders (\apd, \napt and \xml).

Standard	Folder	Schemas	Contents	Origin
TransXChange	root	**TransXChange_registration.xsd**	Terminal schema for Registrations.	Renamed in 2.0.
	root	**TransXChange_general.xsd**	Terminal schema for General use.	Renamed in 2.0.
	root	**TransXChange_general_delta.xsd**	Terminal schema for Delta use.	New in 2.4
TXC	\txc	TXC_authorities-v2-3.xsd		Modularised in 2.4.
	\txc	TXC_common-v2-5.xsd		Renamed in 2.4.
	\txc	TXC_connectingVehicleJourney-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_deadRun-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_flexibleJourneyPattern-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_frequency-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_garage-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_journey_support-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_journeyPattern_support-v2-3.xsd		Modularised in 2.4.
	\txc	TXC_journeyPatternInterchange-v2-3.xsd		Modularised in 2.4.
	\txc	TXC_journeySection-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_journeyPatternTimingLink-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_journeyPattern-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_operatingDay-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_operational-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_operator-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_passingTimes-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_registration-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_route-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_serviceJourneyGrouping-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_service_parts-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_serviceCalendar-v2-5.xsd		Renamed in 2.5.
	\txc	TXC_servicedOrganisation-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_service-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_utility_times-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_utility_types-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_validity-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_vehicleEquipment-v2-55.xsd		New in 2.4.
	\txc	TXC_vehicleJourney-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_vehicleJourney_support-v2-3.xsd		Modularised in 2.4.
	\txc	TXC_vehicleJourneyInterchange-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_vehicleJourney-v2-5.xsd		Modularised in 2.4.
	\txc	TXC_vehicleType-v2-5.xsd		Modularised in 2.5.
NaPT	**\napt**	NaPT_all.xsd		Modularised in 2.4.
	\napt	NaPT_dates-v2-5.xsd		Modularised in 2.4.
	\napt	NaPT_dayTypes-v2-5.xsd		Modularised in 2.4.
	\napt	NaPT_location-v2-5.xsd		Renamed in 2.4.
	\napt	NaPT_operatorSupport-v2-5.xsd		Modularised in 2.5.
	\napt	NaPT_modes-v2-5.xsd		Modularised in 2.4.
	\napt	NaPT_pointOfInterest-v2-5.xsd		New in 2.5.
	\napt	NaPT_stop-v2-5.xsd		Modularised in 2.4.
	\napt	NaPT_stopAccessibility-v2-5.xsd		New in 2.5.
	\napt	NaPT_stopArea-v2-5.xsd		Modularised in 2.4.
	\napt	NaPT_tariffZone-v2-5.xsd		New in 2.5.
	\napt	NaPT_utility_rights-v2-3.xsd		New in 2.4.
	\napt	NaPT_utility_types-v2-5.xsd		Modularised in 2.4.
	\napt	NaPT_utility_units-v2-4.xsd		Modularised in 2.4.
	\napt	NaPT_utility_xml-v2-1.xsd		Modularised in 2.4.
	\napt	NaPT_versioningAttributes-v2-2.xsd		Modularised in 2.4.
NaPT_netex	**\napt**	NaPT_netex_accessibility-v2-5.xsd		New in 2.5.
	\napt	NaPT_netex_entity_support-v2-5.xsd		New in 2.5.
	\napt	NaPT_netex_equipment-v2-5.xsd		New in 2.5.

	\napt	NaPT_netex_equipmentVehiclePassenger-v2-5.xsd		New in 2.5.
	\napt	NaPT_netex_facility_support-v2-5.xsd		New in 2.5.
	\napt	NaPT_netex_facility_version-v2-5.xsd		New in 2.5.
	\napt	NaPT_netex_localService-v2-5.xsd		New in 2.5.
	\napt	NaPT_netex_travelRights_types-v2-5.xsd		New in 2.5.
NPTG	**\nptg**	NPTG_locality-v2-4.xsd		Modularised in 2.4.
	\nptg	NPTG_locality_support-v2-4.xsd		Modularised in 2.4.
	\nptg	NPTG_administrative-v2.5.xsd		Modularised in 2.4.
	\nptg	NPTG_administrative_support-v2-4.xsd		Modularised in 2.4.
GovTalk	**\apd**	**AddessTypes.xsd**	UK address types	Referenced in 2.0
	\apd	**CommonSimpleTypes.xsd**	UK simple types	Referenced in 2.0
W3C	\xml	**XML.xsd**	Standard definitions of types	Referenced in 2.0

Table 12-1 – TransXChange 2.4 Module Names

12.4 Version Identifiers & Change Tracking

12.4.1 Schema Version Identifier

The *TransXChange* schema has an explicit version attribute on it, as recommended by e-GIF.

- The schema ***id*** is “*TransXChange*”.
- The ***version*** identifier follows the versioning scheme e.g. ”3.0”.

12.4.2 Indicating Versions on Data

In each XML instance document conforming to *TransXChange,* the root ***TransXChange*** element has an attribute that is populated to indicate the schema version, as recommended by e-GIF. This allows any application which processes the document to decide how to handle the document. See *Table 12-2*. The Schema version is one of a standard set of Content change attributes that are specified on the route elements of all NaPT schemas.

Attributes		Value Type	Description
CreationDateTime	R	Date and Time stamp, ISO format	When document was first created
ModificationDateTime	R	Date and Time stamp, ISO format	When document was last updated, e.g. to change a workflow attribute.
Modification	R	Nature of modification: one of *new, delete, revise, delta*	Nature of changes in document. Normally revise (New and changes)
RevisionNumber	R	Monotonically incrementing number	Sequentially incrementing number. May be populated by systems that track individual export sessions.
SchemaVersion	R	Schema Version number	Version number of document
ChangesSince	O	Date and Time stamp, ISO format	For deltas only,

Table 12-2 – TransXChange Document Version Attributes

12.4.3 Data Element Version

Most significant entities in *TransXChange* have an optional set of a standard change attributes on them, including a modification date and revision number that can be used to specify their version level. See *Table 12-3*. These can be used by those wishing to support fine-grained round trip version management of individual entities. Fine grained versioning is not required for EBSR.

Change Attributes	Value Type	Description
CreationDateTime	Date and Time stamp, ISO format	Timestamp at creation of entity. Should be set when the entity is first created, and not subsequently be changed.
ModificationDateTime	Date and Time stamp, ISO format	Timestamp at most recent update. Should be changed every time an entity is changed, or when any of its child entities that are not themselves versioned are changed. May be omitted if Modification is new, i.e. if same as CreationDateTime, otherwise must be specified. Will be equal or later than the CreationDateTime .
Modification	Nature of modification: one of n*ew, delete, revise, delta*	Nature of data change of exchanged entity: • *New* : This is the first version of the element instance, as created for the first time. An entity continues to have a status of new until it is revised. The creation date can be used to detect a recent addition. • *Revise*: This is an update to an existing element instance, or any of its child elements are being updated, added, or deleted. Once an element is marked as revise it will continue to be so unless it is marked as deleted, i.e. should not ever revert to new. If no value is specified, revise will be assumed. • *Delete*: The element is being rendered inactive. Records marked as deleted should continue to be exported in subsequent data exchanges but is deprecated against further use. • *Archive*: The element is archived. It will be held in the central database and the identifiers reserved (E.g. Both AtcoCode and NaptanCode), but will be excluded from normal exports. • *Delta*: The element is only a delta: It contains only changes to previous values (mandatory values are always included). Any child elements may also be incomplete and contain only those instances which have changed.

RevisionNumber	Monotonically incrementing number	The RevisionNumber. The revision number of an instance should be incremented (and its Modification value set to 'revised'), if any of its element values, attribute values or contained values are modified by the Originating system. • New entities should have a revision number of 0. • Only the originator of the data should increment this number The RevisionNumber of an instance should not be changed if there is no change to the data values or children of an element.

Table 12-3 – Entity Change Tracking Attributes

- Timestamps should be in standard ISO format, for example *'2004-04-14T14:20:00-05:00"*
- The ***RevisionNumber*** of an element should be incremented (and its ***Modification*** value be set to of *'revised'),* if any of its element values, attribute values or contained values is modified. It may be set to zero for a new entity.

Figure 12-3 shows the common TransXChange versioning attributes

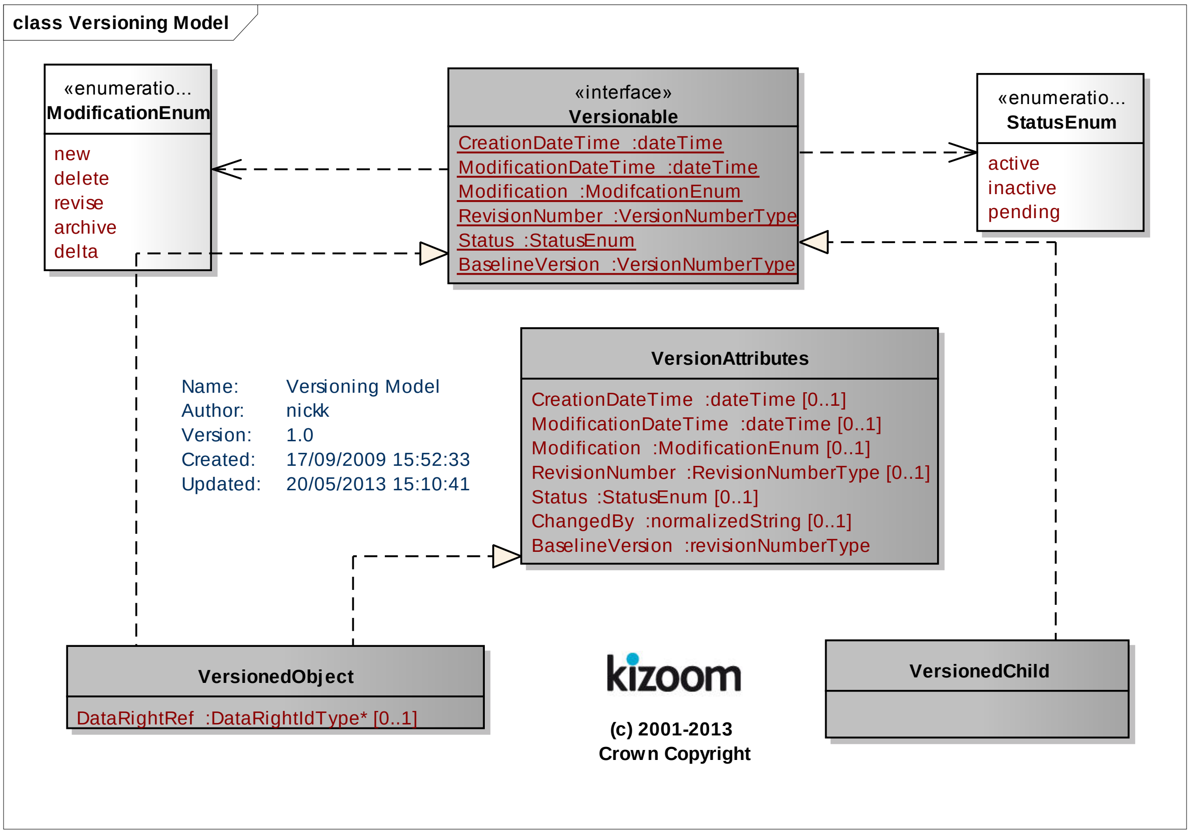


Figure 12-3 – UML diagram of Version Attributes

12.4.4 Change Trackable Entities

The *TransXChange* entities which can be change tracked are shown in *Table 12-4*.

Entity	Versioning	TXC
TransXChange	SchemaVersion + **Document Change Attributes**.	2.1
StopPoint	**Change Attributes.**	2.1
StopArea	**Change Attributes.**	2.1
NptgLocality	**Change Attributes.**	2.1
FlexibleZone	**Change Attributes.**	2.1
Route	**Change Attributes**	2.1
RouteSection	**Change Attributes**	2.1
RouteLink	**Change Attributes**	2.1
Track	**Change Attributes.**	2.4
JourneyPattern	**Change Attributes**	2.1
JourneyPatternSection	**Change Attributes**	2.1
JourneyPatternStopUsage	No – within ***JourneyPatternSection.***	2.1
JourneyPatternTimingLink	**Change Attributes.**	2.4
JourneyPatternInterchange	**Change Attributes.**	2.1
Operator	**Change Attributes.**	2.1
LicencedOperator	**Change Attributes.**	2.1
Garage	**Change Attributes.**	2.4
Service	**Change Attributes.**	2.1
JourneyGrouping	**Change Attributes.**	2.4
Registration	**Change Attributes.**	2.1
Line	**Change Attributes.**	2.4
VehicleJourney	**Change Attributes.**	2.1
ConnectingVehicleJourney	**Change Attributes.**	2.4
FlexibleVehicleJourney	**Change Attributes.**	2.1
VehicleJourneyTimingLink	**Change Attributes.**	2.4
VehicleJourneyStopUsage	No – No within ***VehicleJourney.***	
VehicleJourneyInterchange	**Change Attributes.**	2.1
DeadRun	**Change Attributes.**	2.4
PositioningLink	**Change Attributes.**	2.4
LayoverPoint	**Change Attributes.**	2.4
ServicedOrganisation	**Change Attributes.**	2.1
ServiceFacilitySet	**Change Attributes.**	2.5
VehicleEquipment	**Change Attributes.**	2.4
DayType	**Change Attributes.**	2.4
SupportingDocument	No	2.1
Calendar	**Change Attributes.**	2.4
StopValidity	**Change Attributes.**	2.1
OperatorLicence	**Change Attributes.**	2.4
Descriptor	**Change Attributes.**	2.1
FlexibleServiceTimes	**Change Attributes.**	2.4
AnnotatedCrossRef	**Change Attributes.**	2.4
VariableStopAllocation	**Change Attributes.**	2.4

Table 12-4 – TransXChange Tracked Data Elements

12.5 Exchange of Deltas

EBSR requires the submission of complete coherent schedules using the TransXChange Registration Schema. The schema includes integrity constraints to check that referenced elements are present. The TransXChange General Schema may be used to exchange coherent subsets of a Schedule, as well as lists of stops, serviced Organisations, etc. It likewise includes some referential integrity checks.

Other applications may wish to exchange just the changes to a particular schedule or set of elements. This may be done using the TransXChange Delta schema. This is a special version of the TransXChange schema that has no referential integrity constraints. It is possible to mark whether the set of elements of a specific type is partial or complete using the modification attribute, which may take the values '*delta'* (changes only) '*revise'* (all current & new instances), or '*new'* (new instances only. The delta.xml example provides an example of a delta exchange.

12.6 Names of TransXChange Files

When dealing with a large number of bus schedules, it is helpful for document management if the file name used for a bus schedule when it is exchanged as an XML document gives an indication of its contents. The following format is recommended for file names of *TransXChange* XML documents:

Line_Operator_Area_ServiceCode_StartDate.xml

Where:

- *Line* is the service number seen by the public, as defined by a ***Service / Lines/ Line*** element within the document. If there is more than one ***Line*** associated with a service, use the first.
- *Operator* identifies the service operator and is either:
  - The operator code, i.e. ***RegisteredOperator*** / ***OperatorCode*** element for the service specified the within the document.
  - The operator license number, i.e. ***Operator*** / ***LicenceNumber*** of the operator registering the service.
- *Area* identifies the service area and is either:
  - *Area Code:* Three digit ATCO database code for the district/authority 450. This is the *NPTG* ***AdministrativeArea*** / ***AtcoAreaCode.***
  - *TAN Code:* Two character TAN prefix. This is the ***Registration*** / ***VosaRegistrationNumber / TanCode*** specified the within the document.
- *ServiceCode*- is an arbitrary unique identifier for the service as specified by a ***Service / ServiceCode*** element within the document.
- *StartDate*: Is the registered start date of the service as defined by a ***Service /OperatingPeriod /StartDate*** within the document.

So for example, the *757* service operated by *Aztecbird* (*AZT*) in *West Yorks* (450), the general TransXChange export file name would be:

Using the operator code:
757_AZT_450_4431_20020428.xml

Using the operator licence number:
757_3888_450_4431_20020428.xml

Using the Tan prefix on a registration:
757_3888_PB_4431_20020428.xml

For registrations there should generally be a separate file for each registration change date, i.e. one file for the initial service, one for a new version of the service starting 01/07/2004 and so on.

When exchanging between the authority databases and journey planner and real time systems, multiple services may be contained in a single file, using the general schema.
In this case there is no preferred naming scheming.

13 TRANSMODEL & TRANSXCHANGE COMPARISON

13.1 Transmodel Principles

TransXChange is based on *Transmodel*, a general abstract model for describing public transport information systems, devised on carefully elaborated informational science principles. Some of the key principles for Transmodel may be summarised as follows:

1. **Layered Semantic Models**: The efficient modelling of public transport information requires a number of distinct models, representing different levels of discourse. For example, (i) the geospatial location (i.e. map) layer, (ii) the network topology layer, (iii) the service pattern layer, (iv) the timed vehicle journey layer, (v) the operational running layer, etc.
2. **Projection**: It should be possibly to combine the different models in order to compute over them, relating the corresponding elements of different levels of discourse precisely and unambiguously, using a common frame of reference. For example, route links should map onto geospatial objects such as roads; timing links should map onto route links, etc. The establishment of equivalences between distinct model layers is termed *projection*.
3. **Common Terminology:** A standard set of common conceptual entities should be used for the elements making up the models at each different layer, and a standard *Transmodel* terminology should be used. For example, Line, Journey Pattern, Vehicle Journey, Location.
4. **Point and Link Structures**: Public Transport Information System models typically involve complex networks which are modelled in computer systems by graphs; that is, as networks of nodes (points) and edges (links). Depending on the information of interest in a particular application, it may be appropriate to use ordered collections of links, ordered collections of points, or combinations thereof. Links of a given type should only connect to points of the corresponding semantic level of discourse. Only one unambiguous sequence of points (whether modelled as a point sequence, or link sequence) may be used in a given journey or service pattern.
5. **Well-defined Data Systems**. Elements corresponding to external entities should be assigned unique identifiers from agreed data reference systems.

13.2 Transmodel Terminology

Wherever possible, *TransXChange* follows *Transmodel* terminology for PT concepts. The equivalences between some key *TransXChange* elements and their corresponding *Transmodel* concepts are shown in *Table 13-1*. Divergences are highlighted in bold.

Transmodel	***TransXChange 2.0***	***Previously 1.2***
ADMINSTRATVIE ZONE / AREA	Administrative Area	--
ACTIVITY	Activity	ActivityFlag
Alight	**Set down**	**Set down**
Board	**Pick up**	**Pick up**
BLOCK	Block	--
DAY TYPE	OperatingProfile / RegularDayType PeriodicDayType ServicedOrganisationDayType	DayType / GeneralOpClassification Periodic SchoolOp
DEAD RUN	DeadRun	--
DESTINATION DISPLAY	DestinationDisplay	DynamicDestinationDisplay
DISTANCE	Distance	Distance
DIRECTION	Direction	JourneyDirection
FARE STAGE	FareStage	--
FARE ZONE	FareZone	--
JOURNEY PATTERN	JourneyPattern	JourneyPattern
JOURNEY PATTERN LINK IN SEUENCE + TIMING LINK	JourneyPatternTimingLink	JourneyPatternTimingLink
JOURNEY PATTERN LAYOVER	JourneyPattern / Layover	
JOURNEY PATTERN RUN TIME	JourneyPattern / RunTime	DefaultRunTime
JOURNEY PATTERN WAIT TIME	JourneyPattern / WaitTime	DefaultWaitTime
LINE	Line	(ServiceId)
(FARE SECTION) (LINK SEQUENCE)	RouteSection	--
LOCATION	Location	**Geocode**
LOCATING SYSTEM	LocatingSystem	(Geodata system)
OPERATOR	Operator	Operator
PLACE	Place	**Locality**
ROUTE	**Track (**See 13.3.1)	--
ROUTE LINK	**TrackLink (**See 13.3.1)	--
RUN TIME	Run time	Run time
Section	Section	**--**
SERVICE	Service, StandardService FlexibleService	**OverallServiceDescription**
SERVICE PATTERN	**Route (**See 13.3.1	**Route**
SERVICE JOURNEY PATTERN LINK	**RouteLink (**See 13.3.1	**RouteLink**
SERVICE JOURNEY PATTERN INTERCHANGE	JourneyPatternInterchange	JourneyPatternInterchange
SERVICE JOURNEY INTERCHANGE	VehicleJourneyInterchange	VehicleJourneyInterchange
SITE	**Landmark**	**Landmark**
STOP POINT	StopPoint	**Stop**
STOP AREA	StopArea	**StopCluster**
TIMING LINK	(JourneyPattern) TimingLink	TimingLink
TIME DEMAND	TimeDemand	--
VALIDITY PERIOD	ValidityPeriod	ValidityPeriod
VEHICLE JOURNEY	VehicleJourney	VehicleJourney
(VEHICLE JOURNEY + JOURNEY PATTERN LINK IN SEQUENCE + TIMING LINK)	VehicleJourneyTimingLink	VehicleJourneyTimingLink
VEHICLE JOURNEY RUN TIME	VehicleJourneyTimingLink / RunTime	RunTime
VEHICLE JOURNEY WAIT TIME	VehicleJourneyTimingLink / WaitTime	WaitTime
VEHICLE TYPE	VehicleType	VehicleType
VERSION	**(RevisionNumber)**	**(RevisionNumber)**
WAIT TIME	Wait time	Wait time
ZONE	FlexibleZone	--

Table 13-1 – Comparison of Key Transmodel Terms

13.3 Divergences from Transmodel

Version 2.x of *TransXChange* converges significantly closer to *Transmodel,* but still contains a few significant differences in terminology that reflect *TransXChange* 1.2 usage, and the legacy of *TransXChange's* original ATCO CIF representation. (Note too that *Transmodel* has also been subject to further evolution during the period of development of *TransXChange*). In addition TransXChange introduces additional convenience elements for implementation which can mostly be considered as views which compound other elements; for example a ***StopUsage*** groups various attributes that can be associated with either end of a ***TimingLink***. The main outstanding differences (which may possibly be reduced in future), are as follows:

- *Transmodel* uses the term ROUTE to denote a physical path taken by vehicles through the network, identifying the road sections or track section being used with each stage. A *Transmodel* ROUTE LINK corresponds more properly to the ***Track*** (ROUTE LINK) and ***Mapping*** (POINT IN ROUTE LINK) elements of the *TransXChange* model.

- The *TransXChange* ***Route*** and ***RouteLink*** are similar to a *Transmodel* SERVICE PATTERN, and SERVICE LINK, that is, an abstract journey pattern, identifying a unique sequence of STOP POINTs in order that define a possible journey for a line, regardless of any actual timings.

13.3.1 TransXChange Representation of Journey Patterns

Note that *TransXChange* does not use what *Transmodel* would term a STOP IN SEQUENCE (or more specifically, STOP POINT IN JOURNEY PATTERN) representation of a journey pattern, but rather a *Transmodel* LINKS IN LINK SEQUENCE representation; more specifically, a sequence of journey pattern timing links (TIMING LINK IN JOURNEY PATTERN). The *Transmodel* abstract model allows for a separate set of SERVICE LINKs between the stop points of a service pattern or journey pattern that is distinct from the set of TIMING LINKs of the pattern, permitting multiple timings to be specified for the same route, and for some of the intermediate timing points not to be stop points (and stop points not to be timing points). Because *TransXChange* has historically been primarily concerned with the exchange of fully timed schedules for registration, all points in a *TransXChange* JOURNEY PATTERN are stop points, and *TransXChange* uses only timing links: the existence of a service link between two points is implied from the existence of a timing link between two stops. This simplifies the mapping of the representation to a published matrix timetable; however a consequence is that it forces a false interpolation of run times in some usages. For example, if there is a sequence of non-timing stop points in a pattern, for which there is only an overall run time, the overall run time must be arbitrarily assigned to one or more of the intermediate links in order to encode it in *TransXChange*.

In effect TransXChange makes a simplifying assumption that all TIMING POINTs are in effect also STOP POINTs so is able to use a combined Link abstraction that has both timing and service pattern properties. The StopUsage element

It may be appropriate to add a compatible STOP IN SEQUENCE and separate service and timing link representations to a future version of TransXChange.

13.3.2 Abbreviated Journey Patterns

In *TransXChange,* two practical expedients are used also to reduce the amount of data that has to be exchanged, and in particular the number of journey patterns.

- a. *Short working* of the underlying journey pattern is allowed in *TransXChange*, i.e. truncation of one or more stops of the pattern at either or both ends. *Transmodel* indicates that a separate journey pattern should be declared for *any* difference in stop sequence, which could be strictly interpreted as requiring a separate journey pattern for each short working vehicle journey variant.

- b. *Express* journeys over a service pattern are allowed in *TransXChange* – i.e. provided a journey traverses a link, and goes past a stop, it may specify an activity of ‘pass’ to omit a particular stop.

In both the above cases, there is little or no informational benefit to having a separate journey pattern, and there is in any case little distinction between the above cases and the legitimate *Transmodel* representation of a vehicle following a ‘full’ journey pattern in real-time that for operational reasons passes stops, or terminates early.

13.3.3 Groups of Links

Another expedient *TransXChange,* uses to reduce the amount of data that has to be exchanged is a “link section”, that is, a reusable ordered list of Links that can be reused in one or more ROUTEs or JOURNEY PATTERN. This is particularly useful where there is corridor route with a long common section but many end variants. Link sections are an additional abstraction not found in Transmodel but can be seen as equivalent to GROUP OF LINKs being used in a specific way. Their use amounts to a requirement that there is always at least one “GROUP OF LINKS” associated with each journey pattern. but need not conflict in any way with a canonical Transmodel representation.

14 INTEGRITY RULES

14.1 Syntactic Integrity Rules

XML's inbuilt mechanisms, including *unique* & *keyref* c are used in the *TransXChange* schema to enforce a number of basic integrity checks of data within a *TransXChange* document, including enforcing uniqueness. A document must satisfy these constraints, or it is not well formed and will not be processed further by the *TransXChange Publisher* or other tools.

- Data types are specified for dates, times, durations and other common data types.
- Restricted values are enforced by enumerations – see individual tables of allowed values under the schema guide entry for constrained elements.
- Some additional rules for encoding formatted elements are enforced by regular expressions.
- *Table 14-1* shows the other rules enforced by syntactic constraints.
-

Group	Element / Code	#	Scope	Reference
Code Scope	***StopPoint / AtcoCode***	C1	Stop codes of local ***StopPoint*** & AnnotatedStopPointRef declarations must be unique within document.	***StopPointRef*** instances must reference a ***StopPoint*** or ***AnnotatedStopPoint*** declaration. See also External Integrity rule N1.
	StopArea / StopAreaCode	C2	Codes of local ***StopArea*** (Cluster) declarations must be unique within document.	***StopAreaRef*** instances must reference a ***StopArea***– See External Integrity rule N2.
	ServicedOrganisation / ServicedOrganisationCode	C3	Codes of ***ServicedOrganisation*** declarations must be unique within operator.	***ServicedOrganisationRef*** instances must reference a local definition of a ***ServicedOrganisation*** element. Any ParentServicedOrganisationRef must also be declared
	Service / ServiceCode	C4	***Code*** of each ***Service*** must be unique within document.	***ServiceRef*** instances must refer to a local definition of a ***Service***.
	VehicleJourney / VehicleJourneyCode	C5	***Code*** of each ***VehicleJourney*** & ***FlexibleVehicleJourney*** or ConnectingVehicleJourney must be unique within document.	***VehicleJourneyRef*** instances must reference a local definition of a ***VehicleJourney***.
	Garage / GarageCode	C6	Codes of Garage declarations must be unique within document.	***GarageCodeRef*** instances to a Garage must reference a local definition of a Garage element.
	OperatorCode	C7	Codes of local ***Operator*** declarations must be unique within document.	***OperatorRef*** instances must refer to a local definition of a ***Service***.
Key Uniqueness	***StopPoint / PrivateCode***	U1	***PrivateCodes*** of ***StopPoint*** s must be unique within document	
	StopArea / PrivateCode	U2	***PrivateCodes*** of ***StopAreas*** must be unique within document	
	VehicleJourney / PrivateCode	U3	***PrivateCodes*** of ***VehicleJourneys*** must be unique within document	
	Route / PrivateCode	U4	***PrivateCodes*** of ***Routes*** must be unique within document	
	JourneyPattern / PrivateCode	U5	***PrivateCodes*** of ***JourneyPatterns*** must be unique within document	

	ServicedOrganisation / PrivateCode	U6	***PrivateCodes*** of ***ServicedOrganisation s*** must be unique within document	
	Operator / PrivateCode	U7	***PrivateCodes*** of ***Operator s*** must be unique within document	
	Service / PrivateCode	U8	***PrivateCodes*** of ***Services*** must be unique within document	
	Service / JourneyGrouping	U9	***PrivateCodes*** of ***JourneyGrouping*** must be unique within document (TXC v2.4)	
Identifier Scope	***Route / id***	I1	***id*** of each ***Route*** must be unique within document.	***RouteRef*** instances must reference a local definition of a Route.
	JourneyPattern /id	I2	***id*** of each ***JourneyPattern*** must be unique within document.	***JourneyPatternRef*** instances must reference a local definition of a ***JourneyPattern or FlexibleJourneyPattern***.
	Line / id	I5	***id*** of each ***Line*** must be unique within document.	***LineRef*** instances must refer to a local definition of a ***Line*** element.
	RouteSection / id	I6	***id*** of each ***RouteSection*** must be unique within document.	***RouteSectionRef*** instances must refer to a local definition of a ***RouteSection***.
	JourneyPatternSection / id	I7	***id*** of ***JourneyPatternSection*** must be unique within document.	***JourneyPatternSectionRef*** instances must refer to a local definition of a ***JourneyPatternSection***.
	RouteLink/ id	I8	***id*** of each ***RouteLink*** must be unique within document.	***RouteLinkRef*** instances must reference a local definition of a ***RouteLink***.
	JourneyPatternTimingLink / id	I9	***id*** of each ***JourneyPatternTimingLink*** must be unique within document.	***JourneyPatternRef*** instances must reference a local definition of a ***JourneyPatternTimingLink***.
	VehicleJourneyTimingLink / id	I10	***id*** of each ***VehicleJourneyTimingLink*** must be unique within document.	***VehicleJourneyRef*** instances must reference a local definition of a ***VehicleJourneyTimingLink***.
	JourneyPatternStopUsage / id	I11	***id*** of each ***JourneyPatternStopUsage*** must be unique within document.	***JourneyPatternStopUsageRef*** instances must refer to a local definition of a ***JourneyPatternStopUsage***.
	VehicleJourneyStopUsage / id	I12	***id*** of each ***VehicleJourneyStopUsage*** must be unique within document.	***VehicleJourneyStopUsageRef*** instances must refer to a local definition of a ***VehicleJourneyStopUsage***.
	JourneyPatternInterchange	I13	id of each JourneyPatternInterchange must be unique within document.	JourneyPatternInterchange**Ref** instances must refer to a local definition of a ***VehicleJourneyStopUsage***.
	VehicleJourneyInterchange	I14	id of each ***VehicleJourneyInterchange*** must be unique within document.	***VehicleJourneyInterchange Ref*** instances must refer to a local definition of a ***VehicleJourneyInterchange***.
	DayType / id	I15	***id*** of each ***DayType*** must be unique within document.	***DayTypeRef i***nstances must refer to a local definition of a ***DayType***.
	Operator / id	I16	***id*** of each ***Operator*** must be unique within document.	***OperatorIdRef i***nstances must refer to a local definition of an ***Operator.***
	LicencedOperator / id	I17	***id*** of each ***LicencedOperator*** must be unique within document.	***RegisteredOperatorRef i***nstances must refer to a local definition of an ***Operator.***
	DayType / id	I18	***id*** of each ***DayType*** must be unique within document.	***DayTypeRef i***nstances must refer to a local definition of a ***DayType.***

	Calendar / id	I19	***id*** of each ***Calendar*** must be unique within document.	***Calendar Ref i***nstances must refer to a local definition of a ***Calendar.***
Cyclic	***VehicleJourneyRef***	X1	***VehicleJourney*** must not reference itself.	

Table 14-1 – Syntactic Integrity Rules

14.2 Semantic Integrity Rules

Table 14-3 shows additional integrity rules that need to be applied by applications parsing a *TransXChange* XML document. These are subdivided into two categories:

- *Intrinsic Constraints:* Consistency checks that can be applied without reference to external data. For many of these, a sensible recovery action can be taken.
- *Extrinsic Constraints:* Checks of data values that require reference to an external source*.* Whether these need to be applied depends on the availability of the relevant data sets, and the purpose of the application.

Rules are assigned a severity (see *Table 14-2*) that indicates the likely action that an application such as *TransXChange Publisher* will take if the rule is not satisfied.

Rules that may affect the correct publishing of a document by the *TransXChange* Publisher are marked with a 'p'.

Severity	Meaning	Action
1	Fundamental Inconsistency – Schedule cannot be accurately interpreted.	Report as serious error. Reject for registration.
2	Inconsistency – Default Remedial action possible, but statutory Registration requires clarification.	Report, apply remedy automatically. Reject for registration.
3	Inconsistency – Default Remedial action possible.	Report, apply remedy automatically.
4	Data reference does not exist in external source.	Report as missing.
5	Ancillary data reference does not exist.	Report as missing.
6	Minor data inconsistency.	Report, leave uncorrected.

Table 14-2 – Severity Codes for Semantic Integrity Rules

Group	#	Rule Name	Description	Cat	Sev	Remedy
Metadata	Dc1	Valid FileName	File name is made up of recommended elements.	Int	6	Allow, but give warning.
NaPTAN	Na1	Valid *NaPTAN* Stop Identifiers.	Stop points referenced by an ***AnnotatedStopPointRef*** must exist in the *NaPTAN* database.	Ext	4	Warning.
	Na2	Valid *NaPTAN* StopArea Identifiers.	Stop areas referenced by a ***StopAreaRef*** of a local StopPoint definition must exist in the *NaPTAN* database, or be defined locally.	Ext	4	Warning.
	Na3	Local *NaPTAN* StopAreas.	Stop areas referenced by a ***StopAreaRef*** of a local ***StopPoint*** definition should belong to the same Admin Area as the ***StopPoint*** or to a national area e.g. 910.	Ext	6	Warning.
	Ng3	Valid *NPTG* Localities.	*NPTG* localities referenced by ***NptgLocalityRef*** of local ***StopPoint*** definition must exist in the *NPTG* database.	Ext	4	Warning.
	Ng4	Valid *NPTG* Administrative Areas.	*NPTG* administrative areas referenced by an AdministrativeAreaRef of local stop point definition must exist in the *NPTG* database.	Ext	4	Warning.
Registration	RG1	Justifications	Short term registrations must have at least one justification element (Severity 2 – i.e. required for submission).	Int	3	Warning. (+TXC 2.4)
	RG2	New stops	Only Cancellation may omit new stops required	Int	4	Warning. (+TXC 2.4)
Serviced Organization	Eo1	Valid Serviced Organizations.	For local authorities, should be a valid DfE LEA code. For schools, should be a valid DfE school code.	Ext	5	Warning.

	Eo2	Serviced Organization no cyclic References.	Parent or ancestor should not be self.	Int	3	Ignore parent.
Period	Tp1	Unique Operation Profile weeks.	***PeriodicDayType/ Weeks*** should be distinct.	Int	3,*q	Ignore overlap.
	Tp2	Valid Date Ranges.	***End date*** should be after ***start date*** on ***ValidityPeriod*** and other ranges.	Int	3,*q	Use start date for both, and report.
	Tp3	Distinct Periods.	Periods for exclusion suspension should not overlap.	Int	3	Ignore 2nd period.
	Tp4	Valid Dates	Calendar Dates should lie within Service Operational period	Int	3,*q	Assume within operational period.
Operator	Op1	National Operator Code.	***NationalOperatorCode*** should be valid in future database.	Ext	4	Allow.
	Op2	Distinct Operator References.	***RegisteredOperator*** of a Service should not be the same as A***ssociatedOperator***.	Int	3	Ignore associated operator.
	Op3	Distinct Associated Operator roles.	Each ***AssociatedOperator*** should only be referenced once by a given service for a given role.	Int	6	Ignore duplicate references.
	Op4	Valid garage code.	***GarageCode*** should be valid for Operator.	Int	6	Allow.
Service	Sv1	Flexible Service type.	If ***FlexibleService*** component present, ***ServiceClassification*** should include ***Flexible***.	Int	6	Assume Flexible Type.
	Sv2	Appropriate Service type.	The following combinations of ***ServiceClassification*** are not allowed. • ***NormalStopping*** and any other type except ***RuralService***. • ***ExcursionOrTour*** and any other type.	Int	**2**	DEPRECATED (TXC 2+4)
	Sv3	New stops.	Local stops declared, but registration not flagged as requiring new stops.	Int	3	Assume requires new stops.
	Sv4	Missing Map.	***Service / SchematicMap*** is specified but file not found.	Int	3,*q	Warning.
	Sv5	New short notice application can't exceed change limit.	if ***Service / Application-Classification***> is ***start*** then ***Change-ExceedsLimit*** cannot be true.	Int	3,*q	Warning.
Route	Rs1	Linear routes.	In a sequence of ***RouteSection*** instances making up a given ***Route***, the ***To*** / ***StopPoint*** of the last link of a given ***RouteSection*** should be the same as the ***From*** / ***StopPoint*** of the first link of the succeeding ***RouteSection*** in the ***Route***.	Int	**1,** p	Reject.
	Rs2	Route section link direction.	All route links in a route section should have the same ***Direction.***	Int	6, p	Use first direction found.
	Rs3	Route Direction Antithesis.	For a given ***Service***, any explicit direction values on routes should be an antithetical pair, i.e. Outbound/Inbound, Clockwise/Anticlockwise.	Int	6	Treat Clockwise as Outbound.
	Rl1	Route Link sequence stop references.	In a collection of successive route links, 'To' stop point reference of previous link should be same as 'From' stop reference of next successive link.	Int	3, p	Ignore second usage.
	Rl2	Route Link distinct endpoints.	'From' and 'To' stop points of a ***RouteLink*** should be distinct, i.e. not the same	Int	6	Allow, but issue warning.

	Rl3	Track end points constrained to route.	First and last points of ***Track*** mapping should correspond (i.e. be near to) stop points of parent ***RouteLink***.	Int	3	Ignore points.
	R14	Stop Type Usage	Within a given route Fixed stops (i.e. stops of type MKD) should not fall within the area of Hail and Ride stops (i.e. stops of type HAR)	Int	2	Report as disallowed (TXC+2.4)
Journey Pattern	Jp1	Timing endpoints.	Start and end stops of a journey pattern should have a ***StopType TimingStatus*** of principal point.	Int	4, p	Treat as PTP regardless.
	Jp2	Distinct journey pattern Interchange References.	Inbound and outbound journey patterns at an interchange should normally be distinct.	Int	6	Allow, but give warning.
	Jp3	Journey pattern Direction.	***JourneyPattern*** / ***Direction*** should correspond to one of the ***Service*** direction values. If ***Service*** has only a single direction value, the ***JourneyPattern*** / ***Direction*** should match. If the ***Service / Direction*** has a value of *circular* or *inboundOrOutbound* then JourneyPattern must supply an explicit override rather than using a value of *inherit*?	Int	3	Use Journey Pattern value
	Jps1	Section Projection.	If there are route sections, then for each ***JourneyPatternSection,*** there should be a corresponding ***RouteSection*** with the same number of links.	Int	**1**	Reject.
	Jps2	Linear journey patterns.	In a sequence of ***JourneyPatternSection*** instances making up a given ***JourneyPattern***, the ***To*** / ***StopPoint*** of the last link of a given ***JourneyPatternSection*** should be the same as the ***From*** / ***StopPoint*** of the first link of the succeeding ***JourneyPatternSection*** in the ***JourneyPattern***.	Int	**1**, p	Reject.
	Jptl1	Journey Pattern timing link sequence stop references.	In a collection of successive timing links, 'To' stop reference of previous link should be same as 'From' stop reference of next successive link.	Int	6, p	Ignore second usage.
	Jptl2	Journey Pattern timing link distinct endpoints.	'From' and 'To' stops of a timing link should be distinct, i.e. not the same	Int	6	Allow.
	Jptl3	Route Link Projection.	If a ***JourneyPatternTimingLink*** references a ***RouteLink,*** the start and end stops of both links should correspond. If the ***Direction*** of the ***JourneyPatternTimingLink*** is the same as that of the ***RouteLink***, the respective start points should be the same and the respective ends point should be the same. If the ***Direction*** is opposite, the ***JourneyPatternTimingLink*** start point should match the ***RouteLink*** end point, and vice versa.	Int	**1, p**	Reject
	Jptl4	Start and end activity of journey pattern timing link.	Start activity of first stop of a ***JourneyPattern*** should be pickup only; activity of last stop should be set down. Unless route is circular, or stop connects at a ***JourneyPatternInterchange***.	Int	6	Assume.
	Jptl5	Fare stages consistent with zone numbers.	The ***FareStage*** flag on stop usage of a From stop usage element should be set to reflect any change in ***FareStage*** zone numbers.	Int	6	Assume zone numbers are correct.

	Jptl6	RunTime should be greater than zero.	Only in exceptional cases (e.g. physically adjacent stops) should a timing link run time be zero	Int	6	Allow
Vehicle Journey	Vj1	Cyclic vehicle journey references.	Referenced ***VehicleJourney*** for link usage should not be self, either directly or indirectly.	Int	3, p	Ignore reference.
	Vj2	Vehicle journey link references.	If a ***VehicleJourney*** references a ***VehicleJourney*** for its link usage, there should be no ***VehicleJourneyTimingLink*** instances present for the referencing journey.	Int	3, p	Ignore links in referencing journey.
	Vj3	Mixed Frequency Group	In a group of journeys with the same end making up the same frequent service period, not all vehicle journeys in the group have the same minimum, maximum and scheduled frequencies or minutes past the hour.	Int	3, p	Use values from first (+TXC2.2)
	Vj4	Vehicle journey direction.	Vehicle journey ***Direction*** should be same as the journey pattern ***Direction***.	Int	6	Ignore and use journey pattern value.
	Vj5	Conflicting Frequency Group	In a group of journeys with the same end making up the same frequent service period, either all journeys must use scheduled frequencies or all journeys must use minutes pas the hour. A mixture is not allowed.	Int	3, p	Use values from first (+TXC 2.2)
	Vji1	Distinct interchange references.	Inbound and outbound vehicle journeys of an interchange should be distinct.	Int	3,*q	Allow, but give warning.
	Vji2	Matching interchange journeys.	The vehicle journeys referenced by a ***VehicleJourneyInterchange*** should be dependents of the corresponding inbound and outbound journey patterns referenced by the ***JourneyPatternInterchange*** that the ***VehicleJourneyInterchange*** references.	Int	3	Reject Interchange.
	Vjtl1	Vehicle journey timing link projection.	For each ***VehicleJourneyTimingLink*** there should be a corresponding ***JourneyPatternTimingLink.***	Int	**1**	Reject.
	Vjtl2	Start and end activity of vehicle journey timing link.	Start activity of first stop of a ***VehicleJourney*** should be pickup only; activity of last stop should be set down. Unless route is circular, or stop connects at a ***VehicleJourneyInterchange***.	Int	3,*q	Assume.
	Vjtl3	Short working reference.	Any ***ShortWorking*** / ***JourneyPatternTimingLinkRef*** instances should reference a timing link of the vehicle journey that contains it.	Int	3, p	Ignore short working.
	Vjpl1	Positioning link distinct endpoints.	From and to points of a positioning link should be distinct.	Int	3,*q	Ignore positioning link.
	Vjpl2	Positioning link stop point.	One end of a positioning link sequence should reference a stop in the journey pattern.	Int	3, p	Ignore positioning link sequence.
	Vjpl3	Positioning link reference.	Positioning link references should be valid. Any ***GarageRef*** instances referenced by a positioning link should belong to the ***Service Operator***. Any ***Garage*** Ref, ***LayoverRef*** instances referenced by a positioning link should belong to the ***JourneyPattern.***	Int	3	Ignore positioning link.

Table 14-3 – Intrinsic & Extrinsic Semantic Integrity Rules

14.3 Ordered Relationships

Table 14-4 shows the relationships in *TransXChange* whose order is semantically significant.

From	To	Note
Route	***RouteSection***	Section sequence → Link sequence
RouteSection	***RouteLink***	Route link sequence
JourneyPattern	***JourneyPatternSection***	Section sequence → Link sequence
JourneyPatternSection	***JourneyPatternTimingLink***	Journey Pattern Timing link sequence
VehicleJourney	***VehicleJourneyTimingLink***	Vehicle Journey Timing link sequence
DeadRun	***PositioningLink***	Positioning link sequence
TransXChange	***VehicleJourney***	Journey ordering
RouteLink	***Track***	Track sequence
Track / Mapping	***Location***	Arc of path
Track / Instructions	***Feature***	Steps to traverse track
StopPoint /./ FlexibleZone	***Location***	Bounding box of points
FlexibleJourneyPattern	***FlexibleZones***	Order of visiting zones

Table 14-4 – Ordered Relationships

14.4 Precedence Rules for Combining General Date Elements

Table 14-5 shows the elements governing service dates, in order of precedence. Where elements cover the same day types or date ranges, higher precedence elements are used in preference to lower precedence elements. Data conflicts that are considered validation errors are indicated in a few cases.

Seq	Element		Description	Effect	Error	Sev
1.	Service Period	***Po1***	***Service / OperatingPeriod***			
2.	Vehicle Journey Interchange	***Vi1***	***VehicleJourney / VehicleJourneyInterchange/ ValidityPeriod,***	exclude	T4 Outside of ***Service / OperatingPeriod***	2
3.	Vehicle Journey Special	***Vx1***	***VehicleJourney / OperationProfile / SpecialDaysOperation / DaysOfNonOperation.***	exclude	T4 Outside of ***Service / OperatingPeriod***	2
4.		***Vx2***	***VehicleJourney / OperationProfile / SpecialDaysOperation / DaysOfOperation,***	include	T4 Outside of ***Service / OperatingPeriod***	2
5.		***Vx3***	***VehicleJourney / OperationProfile / BankHolidayOperation / DaysOfNonOperation.***	exclude		
6.		***Vx4***	***VehicleJourney / SpecialOperationProfile / BankHolidayOperation / DaysOfOperation.***	include		
7.	Vehicle Journey Normal	***Vn1***	***VehicleJourney / OperationProfile / ServicedOrganisationDayType / DaysOfNonOperation***	exclude	T4 Outside of ***Service / OperatingPeriod***	2
8.		***Vn2***	***VehicleJourney / OperationProfile / ServicedOrganisationDayType, DaysOfOperation***	include	T4 Outside of ***Service / OperatingPeriod***	2
9.		***Vn3***	***VehicleJourney / OperationProfile / ServicedOrganisationDayType ServicedOrganisation / DaysOfNonOperation*** for the serviced organisations ancestors, as specified by ***ServicedOrganisation / ParentRef.***	exclude		
10.		***Vn4***	***VehicleJourney / OperationProfile / ServicedOrganisationDayType,*** of ***ServicedOrganisation / DaysOfOperation*** for the serviced organisations ancestors, as specified by ***ServicedOrganisation / ParentRef.***	include		
11.		***Vn5***	***VehicleJourney / OperationProfile / PeriodicDayType / WeekOfMonth.***	exclude		
12.		***Vn6***	***VehicleJourney / OperationProfile / RegularDayType / Days.***	include		
13	Journey Pattern Special	***Jx1***	***JourneyPattern / OperationProfile / SpecialDaysOperation / DaysOfNonOperation.***	exclude	T4 Outside of ***Service / OperatingPeriod***	2
14.		***Jx2***	***JourneyPattern / OperationProfile / SpecialDaysOperation / DaysOfOperation,***	include	T4 Outside of ***Service / OperatingPeriod***	2

15.		***Jx3***	***JourneyPattern / OperationProfile / BankHolidayOperation / DaysOfNonOperation.***	exclude		
16.		***Jx4***	***JourneyPattern / SpecialOperationProfile / BankHolidayOperation / DaysOfOperation.***	include		
17.	Journey Pattern Normal	***Jn1***	***JourneyPattern / OperationProfile / ServicedOrganisationDayType / DaysOfNonOperation***	exclude		
18.		***Jn2***	***JourneyPattern / OperationProfile / ServicedOrganisationDayType, DaysOfOperation***	include		
19.		***Jn3***	***JourneyPattern / OperationProfile / ServicedOrganisationDayType ServicedOrganisation / DaysOfNonOperation*** for the serviced organisations ancestors, as specified by ***ServicedOrganisation / ParentRef.***	exclude		
20.		***Jn4***	***JourneyPattern / OperationProfile / ServicedOrganisationDayType,*** of ***ServicedOrganisation / DaysOfOperation*** for the serviced organisations ancestors, as specified by ***ServicedOrganisation / ParentRef.***	include		
21.		***Jn5***	***JourneyPattern / OperationProfile / PeriodicDayType / WeekOfMonth.***	exclude		
22.		***Vn6***	***JourneyPattern / OperationProfile / RegularDayType / Days.***	include		
23.	Journey Pattern Interchange	***Ji1***	***Service / JourneyPatternInterchange / ValidityPeriod,*** outside of range of ***Service / OperatingPeriod***	exclude		
24.	Service Profile	***Sx1***	***Service / SpecialOperationProfile / SpecialDaysOperation / DaysOfNonOperation***	exclude	T4 Outside of ***Service / OperatingPeriod***	2
25.		***Sx2***	***Service / SpecialOperationProfile / SpecialDaysOperation / DaysOfOperation***	include	T4 Outside of ***Service / OperatingPeriod***	2
26.		***Sx3***	***Service / SpecialOperationProfile / BankHolidayOperation / DaysOfNonOperation***	exclude		
27.		***Sx4***	***Service / SpecialOperationProfile / BankHolidayOperation / DaysOfOperation***	include		
28.	Service Normal Profile	***Sn1***	***Service / OperationProfile / ServicedOrganisationDayType / DaysOfNonOperation***	exclude		
29.		***Sn2***	***Service / OperationProfile / ServicedOrganisationDayType, / DaysOfOperation***	include		
30.		***Sn3***	***Service / OperationProfile / ServicedOrganisationDayType,*** of ***ServicedOrganisation / DaysOfNonOperation*** for the serviced organisations ancestors, as specified by ***ServicedOrganisation / ParentRef.***	exclude		
31.		***Sn4***	***Service / OperationProfile / ServicedOrganisationDayType,*** of ***ServicedOrganisation / DaysOfOperation*** for the serviced organisations ancestors, as specified by ***ServicedOrganisation / ParentRef.***	include		
32.		***Sn5***	***Service / OperationProfile / PeriodicDayType / WeekOfMonth.***	exclude		
33.		***Sn6***	***Service / OperationProfile / RegularDayType / Days.***	include		

Table 14-5 – Date Elements in Order of Precedence

15 APPENDIX A – REFERENCES TO OTHER STANDARDS

15.1 Transport Domain

15.1.1 NaPTAN & NPTG

National Public Transport Access Nodes (NaPTAN) database; NaPTAN seeks to assemble and maintain a single source of information on the location and naming of bus stops and other public transport access nodes in England, Wales and Scotland. http://www.traveline.org.uk

	UK Department for Transport NaPTAN & NPTG Schema User Guide 2.5	2013 March	Trapeze
	http://www.dft.gov.uk/naptan/schema/2.5/NaPTAN.xsd	2013 March	Trapeze

15.1.2 JourneyWeb

JourneyWeb is a UK Department for Transport sponsored protocol which defines a national data standard for the dynamic interchange of transport information, including journey plans, and timetables. It is used by the Transport Direct Portal project.

	UK Department for Transport JourneyWeb Schema User Guide 2.5 http://www.dft.gov.uk/journeyweb/schema/2.5/JourneyWeb2.5.xsd	2013 Jan	Trapeze

15.1.3 Transmodel CEN TC 278

Transmodel is a CEN sponsored abstract standard for describing Public Transport Information Systems.

	French Ministry for Transport REFERENCE DATA MODEL FOR PUBLIC TRANSPORT [CEN01] CEN TC278, Reference Data Model For Public Transport, ENV12896 revised, June 2001. [CEN97] CEN TC278, Road Transport and Traffic Telematics - Public Transport -Reference Data Model, ENV 12896, May 1997 http://www.Transmodel.org	2004 Jan	Kizoom

15.1.4 Netex CEN TC 278

Transmodel is a CEN sponsored conceptual model and XML schema based on Transmodel for describing Public Transport Information Systems

	CEN prCEN TC278, NeTEx – Network and Timetable Exchange. prCEN/TS 278308-1:2011 *Part 1: Network Topology* prCEN TC278, NeTEx – Network and Timetable Exchange. prCEN/TS 278308-2:2011 *Part 2: Network Timing Information* *NeTEx —* http://www.netex.org.uk		CEN TC"&* WG3 SG9

15.2 Software & General

15.2.1 XML Schema

http://www.w3.org/XML/Schema

	XML Schema Part 0: Primer http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/	2001 May 2	David C. Fallside
	XML Schema Part 1: Structures http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/	2001 May 2	Various
	XML Schema Part 2: Datatypes http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/	2001 May 2	Paul V. Biron and Ashok Malhotra

15.2.2 ISO Time Formats

	D ISO 8601 Date and Time Formats. http://www.w3.org/TR/xmlschema-2/ – isoformats	2001 May 2	W3C Various

	ISO8601:2000(E) Data elements and interchange formats – Information interchange – Representation of dates and times Second edition 2000-12-15 http://lists.ebxml.org/archives/ebxml-core/200104/pdf00005.pdf	2000 Dec 15	Louis Visser

15.2.3 WGS 1984 Location Referencing

	World Geodetic Standard 1984 https://en.wikipedia.org/wiki/World_Geodetic_System		W3C Various

15.2.4 ISO 639-1 Names of Languages

	ISO 639-1:2001. Code for the representation of the names of languages http://www.oasis-open.org/cover/iso639a.html		Infoterm

15.2.5 Rfc 1766 Tags for the Identification of Languages

	rfc1766 – Tags for the Identification of Languages http://www.ietf.org/rfc/rfc1766.txt		Infoterm

15.2.6 GovTalk XML Coding Standards

GovTalk sets out standards for exchange of data in XML

	Office of the e-Envoy **Schema Guidelines** **Best Practice Advice Version 2** **Schema Guidelines 2.doc**	2002 Oct 12	Paul Spencer
	e-Government Metadata Standard e-GMS1.0 -Government_Metadata_Standard_v1.pdf	2002 Apr	Office of e-Envoy

15.2.7 UML Unified Modelling Language

Unified Modelling Language is a notation for describing software models managed by the Object Management Group.

	Unified Modelling Language (UML), version 1.5 http://www.omg.org/technology/documents/formal/uml.htm	formal/2003-03-01	OMG